

Greedy algorithms for high-dimensional non-symmetric linear problems

Eric Cancès, Virginie Ehrlacher, Tony Lelièvre

October 26, 2012

Abstract

In this article, we present a family of numerical approaches to solve high-dimensional linear non-symmetric problems. The principle of these methods is to approximate a function which depends on a large number of variates by a sum of tensor product functions, each term of which is iteratively computed via a greedy algorithm [20]. There exists a good theoretical framework for these methods in the case of (linear and nonlinear) symmetric elliptic problems. However, the convergence results are not valid any more as soon as the problems considered are not symmetric. We present here a review of the main algorithms proposed in the literature to circumvent this difficulty, together with some new approaches. The theoretical convergence results and the practical implementation of these algorithms are discussed. Their behaviors are illustrated through some numerical examples.

Introduction

High-dimensional problems arise in a wide range of fields such as quantum chemistry, molecular dynamics, uncertainty quantification, polymeric fluids, finance... In all these contexts, one wishes to approximate a function u depending on d variates x_1, \dots, x_d where $d \in \mathbb{N}^*$ is typically very large. Classically, the function u is defined as the solution of a Partial Differential Equation (PDE) and cannot be obtained by standard approximation techniques such as Galerkin methods for instance. Indeed, let us consider a

discretization basis with N degrees of freedom for each variate ($N \in \mathbb{N}^*$), so that the discretization space is given by

$$V_N := \text{Span} \left\{ \psi_{i_1}^{(1)}(x_1) \cdots \psi_{i_d}^{(d)}(x_d), 1 \leq i_1, \dots, i_d \leq N \right\},$$

where for all $1 \leq j \leq d$, $\left(\psi_i^{(j)} \right)_{1 \leq i \leq N}$ is a family of N functions which only depend on the variate x_j . A Galerkin method consists in representing the solution u of the initial PDE as

$$u(x_1, \dots, x_d) \approx \sum_{1 \leq i_1, \dots, i_d \leq N} \lambda_{i_1, \dots, i_d} \psi_{i_1}^{(1)}(x_1) \cdots \psi_{i_d}^{(d)}(x_d),$$

and computing the set of N^d real numbers $(\lambda_{i_1, \dots, i_d})_{1 \leq i_1, \dots, i_d \leq N}$. Thus, the size of the finite-dimensional problem to solve grows exponentially with the number of variates involved in the problem. Such methods cannot be implemented when d is too large: this is the so-called *curse of dimensionality* [2].

Several approaches have recently been proposed in order to circumvent this significant difficulty. Let us mention among others sparse grids [21], tensor formats [11], reduced bases [4] and adaptive polynomial approximations [6].

In this paper, we will focus on a particular kind of methods, originally introduced by Ladevèze *et al.* to do time-space variable separation [12], Chinesta *et al.* to solve high-dimensional Fokker-Planck equations in the context of kinetic models for polymers [1] and Nouy in the context of uncertainty quantification [15], under the name of *Progressive Generalized Decomposition* (PGD) methods.

Let us assume that each variate x_j belongs to a subset \mathcal{X}_j of \mathbb{R}^{m_j} , where $m_j \in \mathbb{N}^*$ for all $1 \leq j \leq d$. For each d -uplet $(r^{(1)}, \dots, r^{(d)})$ of functions such that $r^{(j)}$ only depends on x_j for all $1 \leq j \leq d$, we call a *tensor product function* and denote by $r^{(1)} \otimes \dots \otimes r^{(d)}$ the function which depends on all the variates x_1, \dots, x_d and is defined by

$$r^{(1)} \otimes \dots \otimes r^{(d)} : \begin{cases} \mathcal{X}_1 \times \dots \times \mathcal{X}_d & \rightarrow \mathbb{R} \\ (x_1, \dots, x_d) & \mapsto r^{(1)}(x_1) \cdots r^{(d)}(x_d). \end{cases}$$

The approach of Ladevèze, Chinesta, Nouy and coauthors consists in approximating the function u by a separate variable decomposition, i.e.

$$u(x_1, \dots, x_d) \approx \sum_{k=1}^n r_k^{(1)}(x_1) \cdots r_k^{(d)}(x_d) = \sum_{k=1}^n r_k^{(1)} \otimes \dots \otimes r_k^{(d)}(x_1, \dots, x_d), \quad (1)$$

for some $n \in \mathbb{N}^*$. In the above sum, each term is a tensor product function. Each d -uplet of functions $(r_k^{(1)}, \dots, r_k^{(d)})$ is iteratively computed in a *greedy* [20] way: once the first k terms in the sum (1) have been computed, they are fixed, and the $(k+1)^{th}$ term is obtained as the *next best tensor product function* to approximate the solution. This will be made precise below.

Thus, the algorithm consists in solving several low-dimensional problems whose dimensions scale linearly with the number of variates and may be implementable when classical methods are not. In this case, if we use a discretization basis with N degrees of freedom per variate as above, the size of the discretized problems involved in the computation of a d -uplet $(r_k^{(1)}, \dots, r_k^{(d)})$ scales like Nd and the total size of the discretization problems is nNd .

This numerical strategy has been extensively studied for the resolution of (linear or nonlinear) elliptic problems [20, 13, 10, 5, 18]. More precisely, let u be defined as the unique solution of a minimization problem of the form

$$u = \operatorname{argmin}_{v \in V} \mathcal{E}(v), \quad (2)$$

where V is a reflexive Banach space of functions depending on the d variates x_1, \dots, x_d , and $\mathcal{E} : V \rightarrow \mathbb{R}$ is a coercive real-valued energy functional. Besides, for all $1 \leq j \leq d$, let V_{x_j} be a reflexive Banach space of functions which only depend on the variate x_j . The standard greedy algorithm reads:

1. set $u_0 = 0$ and $n = 1$;
2. find $(r_n^{(1)}, \dots, r_n^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$ such that

$$(r_n^{(1)}, \dots, r_n^{(d)}) \in \operatorname{argmin}_{(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}} \mathcal{E}(u_{n-1} + r^{(1)} \otimes \dots \otimes r^{(d)}),$$

3. set $u_n = u_{n-1} + r_n^{(1)} \otimes \dots \otimes r_n^{(d)}$ and $n = n + 1$.

Under some natural assumptions on the spaces $V, V_{x_1}, \dots, V_{x_d}$ and the energy functional \mathcal{E} , all the iterations of the greedy algorithm are well-defined and the sequence $(u_n)_{n \in \mathbb{N}^*}$ strongly converges in V towards the solution u of the original minimization problem (2).

This result holds in particular when u is defined as the unique solution of

$$\begin{cases} \text{find } u \in V \text{ such that} \\ \forall v \in V, a(u, v) = l(v), \end{cases}$$

where V is a Hilbert space, a a *symmetric* continuous coercive bilinear form on $V \times V$ and l a continuous linear form on V . In this case, u is equivalently solution of a minimization problem of the form (2) with $\mathcal{E}(v) = \frac{1}{2}a(v, v) - l(v)$ for all $v \in V$.

However, when the function u cannot be defined as the solution of a minimization problem of the form (2), designing efficient iterative algorithms is not an obvious task. This situation occurs typically when u is defined as the solution of a *non-symmetric* linear problem

$$\begin{cases} \text{find } u \in V \text{ such that} \\ \forall v \in V, a(u, v) = l(v), \end{cases}$$

where a is a non-symmetric continuous bilinear form on $V \times V$ and l is a continuous linear form on V .

The aim of this article is to give an overview of the state of the art of the numerical methods based on the greedy iterative approach used in this non-symmetric linear context and of the remaining open questions concerning this issue. In Section 1, we present the standard greedy algorithm for the resolution of symmetric coercive high-dimensional problems and the theoretical convergence results proved in this setting. Section 2 explains why a naive transposition of this algorithm for non-symmetric problems is doomed to failure and motivates the need for more subtle approaches. Section 3 describes the certified algorithms existing in the literature for non-symmetric problems. All of them consist in *symmetrizing* the original non-symmetric problem by minimizing the residual of the equation in a well-chosen norm. However, depending on the choice of the norm, either the conditioning of the discretized problems may behave badly or several intermediate problems may have to be solved online, which leads to a significant increase of simulation times and memory needs compared to the original algorithm in a symmetric linear coercive case. So far, there are no methods avoiding these two problems and for which there are theoretical convergence results in the general case. In Section 4, we present some existing algorithms designed by Nouy [16] and Lozinski [14] to circumvent these difficulties and the partial theoretical results which are known for these algorithms. Section 5 is concerned with another algorithm we propose, for which some partial convergence results are proved. In Section 6, the behaviors of the different algorithms presented here are illustrated on simple toy numerical examples. Lastly, we present in the Appendix some possible tracks to design other methods, but for which further work is needed.

1 The symmetric coercive case

1.1 Notation

Let us first introduce some notation. Let d be a positive integer, m_1, \dots, m_d positive integers and $\mathcal{X}_1, \dots, \mathcal{X}_d$ open subsets of $\mathbb{R}^{m_1}, \dots, \mathbb{R}^{m_d}$ respectively.

Let $\mu_{x_1}, \dots, \mu_{x_d}$ denote measures on $\mathcal{X}_1, \dots, \mathcal{X}_d$ respectively. Let $L^2(\mathcal{X}_1; \mu_{x_1}), \dots, L^2(\mathcal{X}_d; \mu_{x_d})$ be associated L^2 spaces, i.e. vectorial spaces which are complete when endowed with the scalar products

$$\forall f, g \in L^2(\mathcal{X}_j; \mu_{x_j}), \quad \langle f, g \rangle_{\mathcal{X}_j} := \int_{\mathcal{X}_j} f(x_j)g(x_j) \mu_{x_j}(dx_j), \quad \forall 1 \leq j \leq d,$$

and their associated norms $\|\cdot\|_{\mathcal{X}_1}, \dots, \|\cdot\|_{\mathcal{X}_d}$. For instance, in the case when $\mathcal{X}_1 = (0, 1)$ and μ_{x_1} is the standard Lebesgue measure on \mathcal{X}_1 , the spaces $L^2(0, 1)$, $L^2_{\text{per}}(0, 1)$ and $L^2_0(0, 1) := \left\{ f \in L^2(0, 1), \int_0^1 f = 0 \right\}$ are examples of such L^2 spaces.

In the rest of this article, for the sake of simplicity, we will omit the reference to the measures $\mu_{x_1}, \dots, \mu_{x_d}$ and denote by $L^2(\mathcal{X}_1) = L^2(\mathcal{X}_1; \mu_{x_1}), \dots, L^2(\mathcal{X}_d) = L^2(\mathcal{X}_d; \mu_{x_d})$.

We introduce the space $L^2(\mathcal{X}_1 \times \dots \times \mathcal{X}_d) := L^2(\mathcal{X}_1) \otimes \dots \otimes L^2(\mathcal{X}_d)$. This space is a Hilbert space when endowed with the natural scalar product

$$\forall f, g \in L^2(\mathcal{X}_1 \times \dots \times \mathcal{X}_d), \quad \langle f, g \rangle := \int_{\mathcal{X}_1 \times \dots \times \mathcal{X}_d} f(x_1, \dots, x_d)g(x_1, \dots, x_d) \mu_{x_1}(dx_1) \dots \mu_{x_d}(dx_d),$$

and the associated norm is denoted by $\|\cdot\|_{\mathcal{X}_1 \times \dots \times \mathcal{X}_d}$.

Let $V \subset L^2(\mathcal{X}_1 \times \dots \times \mathcal{X}_d)$, $V_{x_1} \subset L^2(\mathcal{X}_1)$, \dots , $V_{x_d} \subset L^2(\mathcal{X}_d)$ be Hilbert spaces endowed respectively with scalar products denoted by $\langle \cdot, \cdot \rangle_V, \langle \cdot, \cdot \rangle_{V_{x_1}}, \dots, \langle \cdot, \cdot \rangle_{V_{x_d}}$ and associated norms $\|\cdot\|_V, \|\cdot\|_{V_{x_1}}, \dots, \|\cdot\|_{V_{x_d}}$.

We define $V', V'_{x_1}, \dots, V'_{x_d}$ as the dual spaces of $V, V_{x_1}, \dots, V_{x_d}$ with respect to the L^2 scalar products $\langle \cdot, \cdot \rangle, \langle \cdot, \cdot \rangle_{\mathcal{X}_1}, \dots, \langle \cdot, \cdot \rangle_{\mathcal{X}_d}$. These dual spaces are endowed with their natural norms $\|\cdot\|_{V'}, \dots$ etc.

Lastly, the Riesz operator $R_V : V \rightarrow V'$ is defined by

$$\forall v, w \in V, \quad \langle v, w \rangle_V = \langle R_V v, w \rangle_{V', V}.$$

It holds in particular that $\|v\|_V = \|R_V v\|_{V'}$. Similar operators $R_{V_{x_1}}, \dots, R_{V_{x_d}}$ are introduced for the spaces V_{x_1}, \dots, V_{x_d} .

For any d -uplet $(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$, we define the *tensor product function* $r^{(1)} \otimes \dots \otimes r^{(d)}$ as follows

$$r^{(1)} \otimes \dots \otimes r^{(d)} : \begin{cases} \mathcal{X}_1 \times \dots \times \mathcal{X}_d & \rightarrow \mathbb{R} \\ (x_1, \dots, x_d) & \mapsto r^{(1)}(x_1) \dots r^{(d)}(x_d). \end{cases}$$

In the particular case when $d = 2$, we shall denote respectively $x_1, \mathcal{X}_1, m_1, V_{x_1}$ by x, \mathcal{X}, m_x, V_x and $x_2, \mathcal{X}_2, m_2, V_{x_2}$ by t, \mathcal{T}, m_t, V_t .

Besides, for any Banach spaces H_1, H_2 , the space of bounded linear operators from H_1 to H_2 will be denoted by $\mathfrak{L}(H_1, H_2)$.

1.2 Theoretical results

We recall here the theoretical framework of the standard greedy algorithm in the coercive symmetric case.

Let us consider the problem

$$\begin{cases} \text{find } u \in V \text{ such that} \\ \forall v \in V, a(u, v) = l(v), \end{cases} \quad (3)$$

where

- $a(\cdot, \cdot)$ is a *symmetric, coercive* continuous bilinear form on $V \times V$;
- l is a continuous linear form on V .

Then, problem (3) is equivalent to the minimization problem

$$u = \operatorname{argmin}_{v \in V} \mathcal{E}(v), \quad (4)$$

where

$$\forall v \in V, \mathcal{E}(v) := \frac{1}{2}a(v, v) - l(v). \quad (5)$$

The greedy algorithm reads:

1. let $u_0 = 0$ and $n = 1$;

2. define $(r_n^{(1)}, \dots, r_n^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$ such that

$$(r_n^{(1)}, \dots, r_n^{(d)}) \in \underset{(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}}{\operatorname{argmin}} \mathcal{E}(u_{n-1} + r^{(1)} \otimes \dots \otimes r^{(d)}); \quad (6)$$

3. define $u_n = u_{n-1} + r_n^{(1)} \otimes \dots \otimes r_n^{(d)}$ and set $n = n + 1$.

Let us denote by

$$\Sigma := \{r^{(1)} \otimes \dots \otimes r^{(d)}, r^{(1)} \in V_{x_1}, \dots, r^{(d)} \in V_{x_d}\} \quad (7)$$

and make the following assumptions:

(A1) $\overline{\operatorname{Span}(\Sigma)}^V = V$;

(A2) Σ is weakly closed in V .

These assumptions are usually satisfied in the case of classical Sobolev spaces [5].

Theorem 1.1. *Assume that (A1) and (A2) are satisfied. Then, for all $n \in \mathbb{N}^*$, there exists at least one solution $(r_n^{(1)}, \dots, r_n^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$ (not necessarily unique) to (6) and any solution satisfies $r_n^{(1)} \otimes \dots \otimes r_n^{(d)} \neq 0$ if and only if $u_{n-1} \neq u$. Besides, the sequence $(u_n)_{n \in \mathbb{N}^*}$ strongly converges towards u in V .*

The following Lemma will be used later. Although the proof is given in [10], we recall it here for the sake of self-containedness.

Lemma 1.1. *For all $v \in V$, let us denote by $\|v\|_a := \sqrt{a(v, v)}$. Then, for all $n \in \mathbb{N}^*$,*

$$\|r_n^{(1)} \otimes \dots \otimes r_n^{(d)}\|_a = \sup_{(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}, r^{(1)} \otimes \dots \otimes r^{(d)} \neq 0} \frac{a(u - u_{n-1}, r^{(1)} \otimes \dots \otimes r^{(d)})}{\|r^{(1)} \otimes \dots \otimes r^{(d)}\|_a}. \quad (8)$$

Proof. Let us prove (8) for $n = 1$. The proof is similar for larger $n \in \mathbb{N}^*$. The d -tuple $(r_1^{(1)}, \dots, r_1^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$ solution of (6) for $n = 1$ equivalently satisfies:

$$(r_1^{(1)}, \dots, r_1^{(d)}) \in \underset{(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}}{\operatorname{argmin}} \frac{1}{2} \|u - r^{(1)} \otimes \dots \otimes r^{(d)}\|_a^2. \quad (9)$$

The Euler equations associated to this minimization problem read: for all $(\delta r^{(1)}, \dots, \delta r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$,

$$\begin{aligned} & a \left(r_1^{(1)} \otimes \dots \otimes r_1^{(d)}, r_1^{(1)} \otimes \dots \otimes r_1^{(d-1)} \otimes \delta r^{(d)} + r_1^{(1)} \otimes \dots \otimes r_1^{(d-2)} \otimes \delta r^{(d-1)} \otimes r_1^{(d)} + \dots + \delta r^{(1)} \otimes r_1^{(2)} \otimes \dots \otimes r_1^{(d)} \right) \\ &= a \left(u, r_1^{(1)} \otimes \dots \otimes r_1^{(d-1)} \otimes \delta r^{(d)} + r_1^{(1)} \otimes \dots \otimes r_1^{(d-2)} \otimes \delta r^{(d-1)} \otimes r_1^{(d)} + \dots + \delta r^{(1)} \otimes r_1^{(2)} \otimes \dots \otimes r_1^{(d)} \right) \end{aligned}$$

which implies that

$$\left\| r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right\|_a^2 = a \left(u, r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right). \quad (10)$$

Let now $(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$ be such that $r^{(1)} \otimes \dots \otimes r^{(d)} \neq 0$. Using (9) and (10), it holds that

$$\left\| u - \frac{a \left(u, r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right)}{\left\| r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right\|_a^2} r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right\|_a^2 = \left\| u - r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right\|_a^2 \leq \left\| u - \frac{a \left(u, r^{(1)} \otimes \dots \otimes r^{(d)} \right)}{\left\| r^{(1)} \otimes \dots \otimes r^{(d)} \right\|_a^2} r^{(1)} \otimes \dots \otimes r^{(d)} \right\|_a^2$$

Therefore,

$$\frac{a \left(u, r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right)^2}{\left\| r_1^{(1)} \otimes \dots \otimes r_1^{(d)} \right\|_a^2} \geq \frac{a \left(u, r^{(1)} \otimes \dots \otimes r^{(d)} \right)^2}{\left\| r^{(1)} \otimes \dots \otimes r^{(d)} \right\|_a^2}.$$

Taking the supremum over all $(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}$ such that $r^{(1)} \otimes \dots \otimes r^{(d)} \neq 0$ yields the result. \square

Equation (8) implies in particular that for all $n \in \mathbb{N}^*$,

$$\left\| r_n^{(1)} \otimes \dots \otimes r_n^{(d)} \right\|_a = \sup_{(r^{(1)}, \dots, r^{(d)}) \in V_{x_1} \times \dots \times V_{x_d}} \frac{l \left(r^{(1)} \otimes \dots \otimes r^{(d)} \right) - a \left(u_{n-1}, r^{(1)} \otimes \dots \otimes r^{(d)} \right)}{\left\| r^{(1)} \otimes \dots \otimes r^{(d)} \right\|_a}. \quad (11)$$

Let us rewrite the greedy algorithm in the particular case when $d = 2$.

1. Let $u_0 = 0$ and $n = 1$;

2. define $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \mathcal{E}(u_{n-1} + r \otimes s); \quad (12)$$

3. define $u_n = u_{n-1} + r_n \otimes s_n$ and set $n = n + 1$.

For the sake of simplicity, in the rest of the article, all the algorithms will be presented in the case when $d = 2$. The generalization of the approaches to a larger number of variates d is straightforward unless mentioned.

The Euler equations associated to the minimization problem (12) read

$$a(u_{n-1} + r_n \otimes s_n, \delta r \otimes s_n + r_n \otimes \delta s) = l(\delta r \otimes s_n + r_n \otimes \delta s), \quad \forall (\delta r, \delta s) \in V_x \times V_t. \quad (13)$$

As a consequence of Theorem 1.1, provided that the set

$$\Sigma = \{r \otimes s, r \in V_x, s \in V_t\} \quad (14)$$

satisfies assumptions (A1) and (A2), at the first iteration of the algorithm ($n = 1$), as soon as the form l is nonzero, there exists at least one solution $(r_1, s_1) \in V_x \times V_t$ of

$$a(r_1 \otimes s_1, \delta r \otimes s_1 + r_1 \otimes \delta s) = l(\delta r \otimes s_1 + r_1 \otimes \delta s), \quad \forall (\delta r, \delta s) \in V_x \times V_t,$$

such that $r_1 \otimes s_1 \neq 0$.

In practice, at each iteration $n \in \mathbb{N}^*$, a pair $(r_n, s_n) \in V_x \times V_t$ is computed via the resolution of the Euler equations (13) using a fixed-point procedure which reads as follows:

- choose $(r_n^{(0)}, s_n^{(0)}) \in V_x \times V_t$ and set $m = 1$;

- find $(r_n^{(m)}, s_n^{(m)}) \in V_x \times V_t$ such that

$$\begin{cases} a(u_{n-1} + r_n^{(m)} \otimes s_n^{(m-1)}, \delta r \otimes s_n^{(m-1)}) = l(\delta r \otimes s_n^{(m-1)}), \quad \forall \delta r \in V_x, \\ a(u_{n-1} + r_n^{(m)} \otimes s_n^{(m)}, r_n^{(m)} \otimes \delta s) = l(r_n^{(m)} \otimes \delta s), \quad \forall \delta s \in V_t; \end{cases} \quad (15)$$

- set $m = m + 1$.

This fixed-point algorithm is numerically observed to converge exponentially fast in most situations, although, at least to our knowledge, there is no rigorous proof in the general case.

2 The non-symmetric case

2.1 General framework

Let us now consider the case of a non-symmetric linear problem of the form

$$\begin{cases} \text{find } u \in V \text{ such that} \\ \forall v \in V, a(u, v) = l(v), \end{cases} \quad (16)$$

where

- $a(\cdot, \cdot)$ is a **nonsymmetric** continuous bilinear form on $V \times V$;
- l is a continuous linear form on V .

In the rest of the article, we will assume that

(A3) problem (16) has a unique solution $u \in V$ for any continuous linear form $l \in \mathfrak{L}(V, \mathbb{R})$.

We denote by $\mathcal{A} \in \mathfrak{L}(V, V)$ the operator defined by

$$\forall v, w \in V, \langle \mathcal{A}v, w \rangle_V = a(v, w),$$

and by \mathcal{L} the element of V such that

$$\forall v \in V, \langle \mathcal{L}, v \rangle_V = l(v).$$

We also introduce the operator $A : V \rightarrow V'$ and the linear form $L \in V'$ defined by $A = R_V \mathcal{A}$ and $L = R_V \mathcal{L}$ so that the unique solution u to (16) is also the unique solution to the problem

$$\begin{cases} \text{find } u \in V \text{ such that} \\ Au = L \text{ in } V'. \end{cases}$$

It follows from assumption (A3) that \mathcal{A} and A are invertible operators.

2.2 Prototypical examples

Let us present two prototypical examples we will refer to throughout the rest of the paper.

- The first one is

$$\begin{cases} \text{find } u \in H_0^1(\mathcal{X}) \otimes L^2(\mathcal{T}) \text{ such that} \\ -\Delta_x u + b_x \cdot \nabla_x u + u = f \text{ in } \mathcal{D}'(\mathcal{X} \times \mathcal{T}), \end{cases} \quad (17)$$

with $f \in H^{-1}(\mathcal{X}) \otimes L^2(\mathcal{T})$ and $b_x \in \mathbb{R}^{m_x}$. For this problem, $V = H_0^1(\mathcal{X}) \otimes L^2(\mathcal{T})$, $V' = H^{-1}(\mathcal{X}) \otimes L^2(\mathcal{T})$ and

$$\begin{aligned} \forall u, v \in V, \quad a(u, v) &= \int_{\mathcal{X} \times \mathcal{T}} (\nabla_x u \cdot \nabla_x v + v(b_x \cdot \nabla_x u) + uv), \\ \forall v \in V, \quad l(v) &= \int_{\mathcal{T}} \langle f, v \rangle_{H^{-1}(\mathcal{X}), H_0^1(\mathcal{X})}. \end{aligned}$$

In this case, $A = -\Delta_x + b_x \cdot \nabla_x + 1$.

- The second example is

$$\begin{cases} \text{find } u \in H_0^1(\mathcal{X} \times \mathcal{T}) \text{ such that} \\ -\Delta_{x,t} u + b \cdot \nabla_{x,t} u + u = f \text{ in } \mathcal{D}'(\mathcal{X} \times \mathcal{T}), \end{cases} \quad (18)$$

with $f \in H^{-1}(\mathcal{X} \times \mathcal{T})$ and $b = (b_x, b_t) \in \mathbb{R}^{m_x} \times \mathbb{R}^{m_t}$. For this problem, $V = H_0^1(\mathcal{X} \times \mathcal{T})$, $V' = H^{-1}(\mathcal{X}, \mathcal{T})$ and

$$\begin{aligned} \forall u, v \in V, \quad a(u, v) &= \int_{\mathcal{X} \times \mathcal{T}} (\nabla_{x,t} u \cdot \nabla_{x,t} v + v(b \cdot \nabla_{x,t} u) + uv), \\ \forall v \in V, \quad l(v) &= \langle f, v \rangle_{H^{-1}(\mathcal{X} \times \mathcal{T}), H_0^1(\mathcal{X} \times \mathcal{T})}. \end{aligned}$$

In this case, $A = -\Delta_{x,t} + b \cdot \nabla_{x,t} + 1$.

2.3 Failure of the standard greedy algorithm

Problem (16) cannot be written as a minimization problem of the form (4) with an energy functional given by (5). The definition of the greedy algorithm via the minimization problems (6) or (12) cannot therefore be transposed to this case. However, a natural way to define the iterations of a greedy algorithm for the non-symmetric problem (16) is to define iteratively for $n \in \mathbb{N}^*$ the pair $(r_n, s_n) \in V_x \times V_t$ as a solution of the following equation

$$a(u_{n-1} + r_n \otimes s_n, \delta r \otimes s_n + r_n \otimes \delta s) = l(\delta r \otimes s_n + r_n \otimes \delta s), \quad \forall (\delta r, \delta s) \in V_x \times V_t, \quad (19)$$

by analogy with the Euler equations (13). This is the so-called *PGD-Galerkin* algorithm [3].

Actually, there are cases when $l \neq 0$ and any solution $(r_1, s_1) \in V_x \times V_t$ of the first iteration of the algorithm

$$a(r_1 \otimes s_1, \delta r \otimes s_1 + r_1 \otimes \delta s) = l(\delta r \otimes s_1 + r_1 \otimes \delta s), \quad \forall (\delta r, \delta s) \in V_x \times V_t, \quad (20)$$

necessarily satisfies $r_1 \otimes s_1 = 0$. Such an algorithm cannot converge since the approximation $u_n = \sum_{k=1}^n r_k \otimes s_k$ given by the algorithm is equal to 0 for any $n \in \mathbb{N}^*$. Besides, this situation may occur even when the norm of the antisymmetric part of the bilinear form $a(\cdot, \cdot)$ is arbitrarily small.

Let us give an explicit example.

Example 2.1. Let $\mathcal{X} = \mathcal{T} = (-1, 1)$ and μ_x (respectively μ_t) be the Lebesgue measure on \mathcal{X} (respectively on \mathcal{T}). Let $b \in \mathbb{R}$, $V_x = H_{\text{per}}^1(-1, 1)$, $V_t = L^2(-1, 1)$ and $V = V_x \otimes V_t$. Consider the non-symmetric problem (16) with

$$\forall v, w \in V, \quad a(v, w) = \int_{\mathcal{X} \times \mathcal{T}} (\nabla_x v \cdot \nabla_x w + (b \cdot \nabla_x v)w + vw),$$

and

$$\forall v \in V, \quad l(v) = \int_{\mathcal{X} \times \mathcal{T}} f v,$$

with $f \in L_{\text{per}}^2(-1, 1) \otimes L^2(-1, 1)$.

Problem (16) is equivalent to

$$\begin{cases} \text{find } u \in H_{\text{per}}^1(-1, 1) \otimes L^2(-1, 1) \text{ such that} \\ -\Delta_x u + b \nabla_x u + u = f \quad \text{in } \mathcal{D}'(\mathbb{R} \times \mathcal{T}). \end{cases} \quad (21)$$

In this context, equations (20) read

$$\begin{cases} \text{find } (r_1, s_1) \in H_{\text{per}}^1(-1, 1) \times L^2(-1, 1) \text{ such that} \\ \left[\int_{-1}^1 |s_1(t)|^2 dt \right] (-r_1''(x) + b r_1'(x) + r_1(x)) = \int_{-1}^1 f(x, t) s_1(t) dt, \\ \left[\int_{-1}^1 (|r_1'(x)|^2 + |r_1(x)|^2) dx \right] s_1(t) = \int_{-1}^1 f(x, t) r_1(x) dx, \end{cases} \quad (22)$$

since the periodic boundary conditions on r_1 imply that $\int_{-1}^1 r_1(x) r_1'(x) dx = 0$.

Unlike the symmetric case, there exists an infinite set of functions $f \in L^2_{\text{per}}(-1, 1) \otimes L^2(-1, 1)$ such that $f \neq 0$ and any solution $(r_1, s_1) \in V_x \times V_t$ of equations (22) necessarily satisfies $r_1 \otimes s_1 = 0$ for any arbitrarily small value of $|b|$. This is the case for example when $f(x, t) = \phi(x - t)$ for all $(x, t) \in \mathbb{R} \times (-1, 1)$ with $\phi \in L^2_{\text{per}}(-1, 1)$ an odd real-valued function.

Let us argue by contradiction. If $(r_1, s_1) \in V_x \times V_t$ is a solution to (22) such that $r_1 \otimes s_1 \neq 0$, up to some rescaling, we can assume that

$$\int_{-1}^1 |s_1(t)|^2 dt = \int_{-1}^1 (|r_1'(x)|^2 + |r_1(x)|^2) dx = \lambda > 0.$$

Thus, we can rewrite (22) as

$$\begin{aligned} -r_1''(x) + br_1'(x) + r_1(x) &= \frac{1}{\lambda} \int_{-1}^1 f(x, t) s_1(t) dt, \\ s_1(t) &= \frac{1}{\lambda} \int_{-1}^1 f(x, t) r_1(x) dx. \end{aligned}$$

Plugging the second equation into the first one, we obtain

$$-r_1''(x) + br_1'(x) + r_1(x) = \frac{1}{\lambda^2} \int_{-1}^1 \left(\int_{-1}^1 f(x, t) f(y, t) dt \right) r_1(y) dy. \quad (23)$$

Let us denote by $g(x, y) = \int_{-1}^1 f(x, t) f(y, t) dt$ for all $(x, y) \in \mathbb{R}^2$. As ϕ is an odd, 2-periodic function, it holds that

$$\begin{aligned} g(x, y) &= \int_{-1}^1 f(x, t) f(y, t) dt \\ &= \int_{-1}^1 \phi(x - t) \phi(y - t) dt \\ &= - \int_{-1}^1 \phi(x - t) \phi(t - y) dt \\ &= - \int_{-1+y}^{1+y} \phi(x - y - u) \phi(u) du \\ &= - \int_{-1}^1 \phi(x - y - u) \phi(u) du. \end{aligned}$$

Taking the Fourier transform of equation (23) yields that for all $k \in \pi\mathbb{Z}$,

$$(|k|^2 + ibk + 1)\widehat{r}_1(k) = -\frac{4}{\lambda^2} \left(\widehat{\phi}(k)\right)^2 \widehat{r}_1(k),$$

where

$$\widehat{r}_1(k) = \frac{1}{2} \int_{-1}^1 r_1(x) e^{-ik \cdot x} dx.$$

Furthermore, $\lambda \in \mathbb{R}_+^*$ and $\widehat{\phi}(0) = 0$ (ϕ is an odd function). Thus, since $\widehat{\phi}(k)$ is a purely imaginary number, $-\left(\widehat{\phi}(k)\right)^2 = \left|\widehat{\phi}(k)\right|^2$ and a solution r_1 necessarily satisfies $\widehat{r}_1(k) = 0$ for all $k \in \pi\mathbb{Z}$, which yields a contradiction.

This example clearly shows that a naive transposition of the greedy algorithm to the non-symmetric case by analogy with the Euler equations (13) obtained in the symmetric case may be doomed to failure.

This article presents a review of some methods which aim at circumventing this difficulty. A particular highlight is set on the practical implementation of these methods and on the existence of theoretical rigorous convergence results. The properties of the different algorithms which are dealt with in this article are summarized in Figure 1.

3 Residual minimization algorithms

In this section, we present some numerical methods used for the computation of separate variable representations of the solution of non-symmetric problems, for which there are rigorous convergence proofs. A natural idea is to symmetrize (16) using a reformulation as a residual minimization problem in a well-chosen norm. These algorithms are also called *Minimum Residual PGD* in the literature [3].

3.1 Minimization of the residual in the $L^2(\mathcal{X} \times \mathcal{T})$ norm

Let us assume that $L \in L^2(\mathcal{X} \times \mathcal{T})$ and that there exists $D(A) \subset V$ a dense subdomain of $L^2(\mathcal{X} \times \mathcal{T})$ such that $A(D(A)) \subset L^2(\mathcal{X} \times \mathcal{T})$. The mapping $A : D(A) \rightarrow L^2(\mathcal{X} \times \mathcal{T})$ defines a linear operator on $L^2(\mathcal{X} \times \mathcal{T})$. Let us assume moreover that A is a closed operator. This implies in particular that $D(A)$, endowed with the scalar product

$$\forall v, w \in D(A), \langle v, w \rangle_{D(A)} = \langle v, w \rangle + \langle Av, Aw \rangle,$$

| | THEORY | PRACTICE |
|---------------------------------|--|---|
| L^2 residual minimization | OK Additional regularity on the right-hand side is needed though. | The conditioning of the resulting problems scale quadratically with the conditioning of the original problem. |
| Dual norm residual minimization | OK | Need to solve several small- or high- dimensional symmetric coercive problems |
| PGD-Galerkin | There are cases where the algorithm does not converge towards the true solution. | OK |
| Minimax | OK for separated operators | OK |
| Dual Greedy | OK in finite dimension when $V = V_x \otimes V_t$ Problems in infinite dimension | OK |
| X-Greedy | Same situation as the Dual-Greedy | Not clear how to implement the algorithm in practice. |
| Decomposition | OK in finite dimension provided that the explicit part of the bilinear form is small enough compared to its implicit part. | OK but slow Diverges if the explicit part of the bilinear form is too large. |

Figure 1: Summary of the different greedy algorithms used for non-symmetric high-dimensional linear problems.

is a Hilbert space.

A first approach, inspired by [9], consists in applying a standard greedy algorithm on the energy functional

$$\mathcal{E}(v) = \|Av - L\|_{L^2(\mathcal{X} \times \mathcal{T})}^2, \quad \forall v \in D(A).$$

Let us consider the case when

$$A = \sum_{i=1}^p A_x^{(i)} \otimes A_t^{(i)}$$

where for all $1 \leq i \leq p$, $A_x^{(i)}$ and $A_t^{(i)}$ are operators on $L^2(\mathcal{X})$ and $L^2(\mathcal{T})$ with domains $D(A_x^{(i)})$ and $D(A_t^{(i)})$ respectively. We denote by $D_x = \bigcap_{i=1}^p D(A_x^{(i)})$ and $D_t = \bigcap_{i=1}^p D(A_t^{(i)})$, and assume that D_x and D_t are dense subspaces of $L^2(\mathcal{X})$ and $L^2(\mathcal{T})$ respectively and are Hilbert spaces, when endowed with the scalar products

$$\forall v, w \in D_x, \langle v, w \rangle_{D_x} = \langle v, w \rangle_{\mathcal{X}} + \sum_{i=1}^p \langle A_x^{(i)} v, A_x^{(i)} w \rangle_{\mathcal{X}},$$

and

$$\forall v, w \in D_t, \langle v, w \rangle_{D_t} = \langle v, w \rangle_{\mathcal{T}} + \sum_{i=1}^p \langle A_t^{(i)} v, A_t^{(i)} w \rangle_{\mathcal{T}}.$$

The greedy algorithm reads:

1. let $u_0 = 0$ and set $n = 1$;
2. define $(r_n, s_n) \in D_x \times D_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in D_x \times D_t}{\operatorname{argmin}} \|A(u_{n-1} + r \otimes s) - L\|_{L^2(\mathcal{X} \times \mathcal{T})}^2; \quad (24)$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$.

Let us denote by $\Sigma^D := \{r \otimes s, r \in D_x, s \in D_t\}$. From Theorem 1.1, provided that

$$(B1) \quad \overline{\operatorname{Span} \Sigma^D}^{D(A)} = D(A);$$

(B2) Σ^D is weakly closed in $D(A)$;

the sequence $(u_n)_{n \in \mathbb{N}^*}$ strongly converges towards u in $D(A)$.

In the case of problem (17), $A = A_x \otimes A_t$ with $A_x = -\Delta_x + b \cdot \nabla_x + 1$ and $A_t = 1$, $D(A) = (H^2(\mathcal{X}) \cap H_0^1(\mathcal{X})) \otimes L^2(\mathcal{T})$, $D_x = D(A_x) = H^2(\mathcal{X}) \cap H_0^1(\mathcal{X})$ and $D_t = D(A_t) = L^2(\mathcal{T})$.

For problem (18), $A = A_x^{(1)} \otimes A_t^{(1)} + A_x^{(2)} \otimes A_t^{(2)}$ with $A_x^{(1)} = -\Delta_x + b_x \cdot \nabla_x + 1$, $A_t^{(1)} = 1$, $A_x^{(2)} = 1$ and $A_t^{(2)} = -\Delta_t + b_t \cdot \nabla_t$, $D(A) = H^2(\mathcal{X} \times \mathcal{T}) \cap H_0^1(\mathcal{X} \times \mathcal{T})$, $D_x = H^2(\mathcal{X}) \cap H_0^1(\mathcal{X})$ and $D_t = H^2(\mathcal{T}) \cap H_0^1(\mathcal{T})$.

In both cases, assumptions (A1) and (A2) are satisfied.

Actually, when L is regular enough, i.e. if $L \in D(A^*)$, where A^* denotes the adjoint of A and $D(A^*)$ its domain, this method is equivalent to performing a standard greedy algorithm on the symmetric coercive problem

$$A^* A u = A^* L.$$

The Euler equations associated to the minimization problems (24) read

$$\langle A(u_{n-1} + r_n \otimes s_n) - L, A(\delta r \otimes s_n + r_n \otimes \delta s) \rangle = 0, \quad \forall (\delta r, \delta s) \in D_x \times D_t.$$

This method suffers from several drawbacks though. Firstly, the right-hand side L needs more regularity than necessary for problem (16) to be well-posed (we need $L \in L^2(\mathcal{X} \times \mathcal{T})$ instead of $L \in V'$).

Secondly, and more importantly, the conditioning of the associated discretized problems behaves badly since it scales quadratically with the conditioning of the original problem $Au = L$.

3.2 Minimization of the residual in the dual norm

In order to avoid the conditioning problems encountered when minimizing the residual in the $L^2(\mathcal{X} \times \mathcal{T})$ norm, another method consists in performing a greedy algorithm on the energy functional

$$\mathcal{E}(v) = \|Av - L\|_{V'}^2 = \|R_V^{-1}(Av - L)\|_V^2, \quad \forall v \in V.$$

Here, the residual $Av - L$ is evaluated in the dual norm $\|\cdot\|_{V'}$. In this method, the right-hand side L does not need to be more regular than $L \in V'$ and this

approach is equivalent to performing the standard greedy algorithm on the symmetric coercive problem

$$A^*(R_V)^{-1}Au = A^*(R_V)^{-1}L.$$

The conditioning of the resulting problem scales linearly with the conditioning of the original $Au = L$ problem.

The algorithm reads:

1. let $u_0 = 0$ and $n = 1$;
2. let $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \| (R_V)^{-1} [A(u_{n-1} + r \otimes s) - L] \|_V^2; \quad (25)$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$.

Provided that Σ defined by (14) satisfies assumptions (A1) and (A2), we infer from Theorem 1.1 that the sequence $(u_n)_{n \in \mathbb{N}}$ strongly converges to u in V .

The Euler equations associated with the minimization problems (25) read: for all $(\delta r, \delta s) \in V_x \times V_t$,

$$\langle R_V^{-1} [A(u_{n-1} + r_n \otimes s_n) - L], R_V^{-1} [A(\delta r \otimes s_n + r_n \otimes \delta s)] \rangle_V = 0,$$

or equivalently,

$$\langle A(u_{n-1} + r_n \otimes s_n) - L, R_V^{-1} [A(\delta r \otimes s_n + r_n \otimes \delta s)] \rangle_{V', V} = 0.$$

However, even if the conditioning problem of the previous method is avoided, this algorithm still requires the inversion of the operator R_V .

In the case when $V = V_x \otimes V_t$, the dual space V' satisfies $V' = V'_x \otimes V'_t$, so that the operator $R_V = R_{V_x} \otimes R_{V_t}$ is a tensorized operator and $R_V^{-1} = R_{V_x}^{-1} \otimes R_{V_t}^{-1}$. A prototypical example of this situation is given in problem (17), where we have $V'_x = H^{-1}(\mathcal{X})$, $V'_t = L^2(\mathcal{T})$, $R_{V_x} = -\Delta_x$, $R_{V_t} = 1$ and $R_V = R_{V_x} \otimes R_{V_t}$. Thus, $R_V^{-1} = (-\Delta_x)^{-1} \otimes 1$ and carrying out the above greedy algorithm requires the computation of several low-dimensional Poisson problems, which remains doable but increases the time and memory needs compared to a standard greedy algorithm in the symmetric coercive case where $b = b_x = 0$.

The situation is even more intricate when $V \neq V_x \otimes V_t$, since the operator R_V is not a tensorized operator in general. A prototypical example of this situation is problem (18) where $V' = H^{-1}(\mathcal{X} \times \mathcal{T})$, $R_V = -\Delta_{x,t}$ and R_V^{-1} cannot be expanded as a finite sum of tensorized operators. These intermediate symmetric coercive high-dimensional can be solved with a standard greedy algorithm presented in Section 1, but this considerably increases the time needed to run a simulation.

In this particular case, since $R_V = -\Delta_x \otimes 1 - 1 \otimes \Delta_t$ is the sum of two tensorized operators which commute with one another, we can use an approach described in [11]. This method consists in using an *approximate expansion* of the inverse of the Laplacian operator, constructed as follows. The function $h : x \in [x_0, +\infty) \mapsto \frac{1}{x}$ (where x_0 is a positive real number) can be approximated by a sum of exponential functions of the form

$$\frac{1}{x} \approx \sum_{l=1}^N C_l e^{-c_l x},$$

for some $N \in \mathbb{N}^*$ and where $(C_l)_{1 \leq l \leq N}$ and $(c_l)_{1 \leq l \leq N}$ are two sets of well-chosen real numbers, depending on x_0 . Provided that x_0 satisfies $x_0 < \min(1, \lambda_1^x, \lambda_1^t)$, where λ_1^x (respectively λ_1^t) is the lowest eigenvalue of the operator $-\Delta_x$ on $H_0^1(\mathcal{X})$ with respect to the $L^2(\mathcal{X})$ scalar product (respectively the lowest eigenvalue of the operator $-\Delta_t$ on $H_0^1(\mathcal{T})$ with respect to the $L^2(\mathcal{T})$ scalar product), since both the operators $-\Delta_x \otimes 1$ and $-1 \otimes \Delta_t$ commute, R_V^{-1} can be approximated by

$$\begin{aligned} R_V^{-1} &\approx \sum_{l=1}^N C_l e^{-c_l (-\Delta_x \otimes 1 - 1 \otimes \Delta_t)} \\ &= \sum_{l=1}^N C_l e^{-c_l \Delta_x} \otimes e^{-c_l \Delta_t}. \end{aligned} \tag{26}$$

The computation of the expansion (26) only involves the computation of the exponential of small-dimensional operators. But of course, to have a reliable approximation of this operator, the number N of terms in the above approximation may be very large. Besides, an explicit expansion is not always available for a general operator R_V^{-1} .

The algorithms presented in the following sections are attempts to find numerical methods which

- avoid the conditioning problem inherent to the method described in Section 3.1;

- avoid the use of inverse operators such as R_V^{-1} in the approach using the dual norm.

Of course, a natural idea would be to find a suitable norm to minimize the residual to avoid the conditioning and inversion problems. So far, no norms with such properties have been proposed.

In Section 4, we present algorithms already existing in the literature, namely those suggested by Anthony Nouy [16] and Alexei Lozinski [14]. In Section 5, a new algorithm is proposed. The known partial convergence results for these methods are presented and the numerical implementation of the algorithms are detailed.

4 Algorithms based on dual formulations

In this section, we present some classes of algorithms based on dual formulations of the non-symmetric problem (16).

4.1 MiniMax algorithm

A first algorithm based on a dual formulation of problem (16) is the *MiniMax algorithm* proposed by Nouy [16].

The algorithm reads as follows:

1. let $u_0 = 0$ and $n = 1$;
2. let $(r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in V_x^2 \times V_t^2$ such that

$$(r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in \arg \max_{(\tilde{r}, \tilde{s}) \in V_x \times V_t} \min_{(r, s) \in V_x \times V_t} \mathcal{J}_n(r \otimes s, \tilde{r} \otimes \tilde{s}), \quad (27)$$

where for all $v, \tilde{v} \in V$,

$$\mathcal{J}_n(v, \tilde{v}) = \frac{1}{2} \|v\|_V^2 - a(u_{n-1} + v, \tilde{v}) + l(\tilde{v});$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$.

At each iteration $n \in \mathbb{N}^*$, the computation of a quadruplet $(r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in V_x^2 \times V_t^2$ satisfying (27) is done by solving the stationarity equations

$$\begin{cases} a(u_{n-1} + r_n \otimes s_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) = l(\tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n), & \forall (\delta \tilde{r}, \delta \tilde{s}) \in V_x \times V_t, \\ a(r_n \otimes \delta s + \delta r \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n) = \langle r_n \otimes \delta s + \delta r \otimes s_n, r_n \otimes s_n \rangle_V, & \forall (\delta r, \delta s) \in V_x \times V_t. \end{cases} \quad (28)$$

In practice, for each $n \in \mathbb{N}^*$, these equations are solved through a fixed-point procedure where the pairs $(r_n, \tilde{r}_n) \in V_x^2$ and $(s_n, \tilde{s}_n) \in V_t^2$ are computed iteratively. More precisely, the fixed-point algorithm reads:

- set $m = 0$, and choose an initial guess $(r_n^{(0)}, \tilde{r}_n^{(0)}, s_n^{(0)}, \tilde{s}_n^{(0)}) \in V_x^2 \times V_t^2$;

- find $(r_n^{(m+1)}, \tilde{r}_n^{(m+1)}) \in V_x^2$ such that

$$\begin{cases} a(u_{n-1} + r_n^{(m+1)} \otimes s_n^{(m)}, \delta \tilde{r} \otimes \tilde{s}_n^{(m)}) = l(\delta \tilde{r} \otimes \tilde{s}_n^{(m)}), & \forall \delta \tilde{r} \in V_x, \\ a(\delta r \otimes s_n^{(m)}, \tilde{r}_n^{(m+1)} \otimes \tilde{s}_n^{(m)}) = \langle \delta r \otimes s_n^{(m)}, r_n^{(m+1)} \otimes s_n^{(m)} \rangle_V, & \forall \delta r \in V_x; \end{cases}$$

- find $(s_n^{(m+1)}, \tilde{s}_n^{(m+1)}) \in V_t^2$ such that

$$\begin{cases} a(u_{n-1} + r_n^{(m+1)} \otimes s_n^{(m+1)}, \tilde{r}_n^{(m+1)} \otimes \delta \tilde{s}) = l(\tilde{r}_n^{(m+1)} \otimes \delta \tilde{s}), & \forall \delta \tilde{s} \in V_t, \\ a(r_n^{(m+1)} \otimes \delta s, \tilde{r}_n^{(m+1)} \otimes \tilde{s}_n^{(m+1)}) = \langle r_n^{(m+1)} \otimes \delta s, r_n^{(m+1)} \otimes s_n^{(m+1)} \rangle_V, & \forall \delta s \in V_t; \end{cases}$$

- set $m = m + 1$.

In [17], it is proved that in the case when $a = a_x \otimes a_t$ where a_x is a continuous bilinear form on $V_x \times V_x$ and a_t a continuous bilinear form on $V_t \times V_t$ and $V = V_x \otimes V_t$, the algorithm converges. However, there is no convergence result in the full general case.

4.2 Greedy algorithms for Banach spaces

Another family of dual greedy algorithms is inspired from the methods suggested by Temlyakov in [20] for Banach spaces and was proposed by Lozinski [14] in order to deal with the resolution of high-dimensional problems of the form (16).

4.2.1 Greedy algorithms for general Banach spaces

For the sake of simplicity, let us present two particular greedy algorithms proposed by Temlyakov in the context of Banach spaces, namely the *X-Greedy* and the *Dual Greedy* algorithms.

Let $(X, \|\cdot\|_X)$ be a reflexive Banach space and \mathcal{D} a dictionary of X , i.e. a subset of X such that for all $g \in \mathcal{D}$, $\|g\|_X = 1$ and $\overline{\text{Span}(\mathcal{D})}^X = X$. Let us also denote by X^* the dual space of X .

Let $f \in X$. The aim of both the Dual Greedy and the X-Greedy algorithms is to give an approximation of f as a linear combination of vectors of the dictionary \mathcal{D} . These numerical methods are generalizations of the *Pure Greedy* algorithm, which is defined for Hilbert spaces. When X is a Hilbert space endowed with the scalar product $\langle \cdot, \cdot \rangle_X$, the Pure Greedy algorithm can be interpreted in two equivalent ways, namely:

Pure Greedy algorithm (1):

1. let $f_0 = 0$, $r_0 = f$ and $n = 1$;
2. let $g_n \in \mathcal{D}$ and $\alpha_n \in \mathbb{R}$ such that (assuming existence)

$$\|r_{n-1} - \alpha_n g_n\|_X = \min_{g \in \mathcal{D}, \alpha \in \mathbb{R}} \|r_{n-1} - \alpha g\|_X;$$

3. let $f_n = f_{n-1} + \alpha_n g_n$, $r_n = r_{n-1} - \alpha_n g_n$ and $n = n + 1$;

and

Pure Greedy algorithm (2):

1. let $f_0 = 0$, $r_0 = f$ and $n = 1$;
2. let $g_n \in \mathcal{D}$ such that (assuming existence)

$$\langle r_{n-1}, g_n \rangle_X = \max_{g \in \mathcal{D}} \langle r_{n-1}, g \rangle_X;$$

3. let $\alpha_n \in \mathbb{R}$ such that

$$\|r_{n-1} - \alpha_n g_n\|_X = \min_{\alpha \in \mathbb{R}} \|r_{n-1} - \alpha g_n\|_X;$$

4. let $f_n = f_{n-1} + \alpha_n g_n$, $r_n = r_{n-1} - \alpha_n g_n$ and $n = n + 1$.

When X is a Hilbert space, the two versions of the Pure Greedy algorithm are equivalent, but this is not the case anymore as soon as X is a general Banach space.

The *X-Greedy* algorithm corresponds to the extension of the first version of the Pure Greedy algorithm:

1. let $f_0 = 0$, $r_0 = f$ and $n = 1$;
2. let $g_n \in \mathcal{D}$ and $\alpha_n \in \mathbb{R}$ such that (assuming existence)

$$\|r_{n-1} - \alpha_n g_n\|_X = \min_{g \in \mathcal{D}, \alpha \in \mathbb{R}} \|r_{n-1} - \alpha g\|_X; \quad (29)$$

3. let $f_n = f_{n-1} + \alpha_n g_n$, $r_n = r_{n-1} - \alpha_n g_n$ and $n = n + 1$.

The *Dual Greedy* algorithm generalizes the second version of the Pure Greedy algorithm and is slightly more subtle. It is based on the notion of *peak functional*. For any non-zero element $f \in X$, we say that $F_f \in X'$ is a peak functional for f if $\|F_f\|_{X^*} = 1$ and $F_f(f) = \|f\|_X$. The *Dual Greedy* algorithm reads:

1. let $f_0 = 0$, $r_0 = f$ and $n = 1$;
2. let $F_{r_{n-1}} \in X^*$ be a peak functional for r_{n-1} and let $g_n \in \mathcal{D}$ such that (assuming existence)

$$g_n \in \operatorname{argmax}_{g \in \mathcal{D}} F_{r_{n-1}}(g); \quad (30)$$

3. let $\alpha_n \in \mathbb{R}$ such that

$$\alpha_n \in \operatorname{argmin}_{\alpha \in \mathbb{R}} \|r_{n-1} - \alpha g_n\|_X; \quad (31)$$

4. let $f_n = f_{n-1} + \alpha_n g_n$, $r_n = r_{n-1} - \alpha_n g_n$ and $n = n + 1$.

Slightly modified versions (relaxed versions) of the X-Greedy and Dual Greedy algorithms are proved to converge in [20] provided that the space X and the dictionary \mathcal{D} satisfy some additional assumptions, detailed below.

We define the modulus of smoothness of the Banach space X by

$$\forall \beta \in \mathbb{R}, \rho(\beta) := \sup_{\|x\|_X = \|y\|_X = 1} \left(\frac{1}{2} (\|x + \beta y\|_X + \|x - \beta y\|_X) - 1 \right).$$

The Banach space $(X, \|\cdot\|_X)$ is said to be uniformly smooth [20] if

$$\lim_{\beta \rightarrow 0} \frac{\rho(\beta)}{\beta} = 0.$$

Let us point out that if a Banach space $(X, \|\cdot\|_X)$ is uniformly smooth, then the mapping $G : x \in X \mapsto \|x\|_X$ is Fréchet-differentiable.

The relaxed versions of the X-Greedy and Dual Greedy algorithms are proved to converge [20] provided that

- (B1) $\overline{\text{Span}(\mathcal{D})}^{\|\cdot\|_X} = X$;
- (B2) $\mathbb{R}\mathcal{D}$ is weakly closed in X ;
- (B3) X is a uniformly smooth Banach space.

We do not write here these relaxed versions of the algorithms for the sake of brevity and refer to [20].

4.2.2 Special Banach spaces for non-symmetric high-dimensional problems

Let us now present how these ideas were adapted by Lozinski to the case of high-dimensional non-symmetric problems. We begin here with the description of the particular Banach spaces involved. Let us assume in the rest of Section 4.2 that the operator $\mathcal{A}^{-1} : V \rightarrow V$ is bounded.

A Banach space with good theoretical properties but which cannot be used in practice

The space V is now endowed with the following dual norm

$$\forall v \in V, \|v\|_{\mathcal{A}} = \sup_{w \in V, w \neq 0} \frac{a(v, w)}{\|w\|_V} = \|\mathcal{A}v\|_V = \|Av\|_{V'}.$$

Actually, since the linear operator \mathcal{A} is bounded on V , the space $(V, \|\cdot\|_{\mathcal{A}})$ is a reflexive Banach space whose dual space is $(V, \|\cdot\|_{(\mathcal{A}^*)^{-1}})$ where

$$\forall v \in V, \|v\|_{(\mathcal{A}^*)^{-1}} = \sup_{w \in V, w \neq 0} \frac{a(w, v)}{\|w\|_V} = \|(\mathcal{A}^*)^{-1}v\|_V.$$

Let us show that the Banach space $(V, \|\cdot\|_{\mathcal{A}})$ and the dictionary

$$\mathcal{D} = \{r \otimes s, r \in V_x, s \in V_t, \|r \otimes s\|_{\mathcal{A}} = 1\}$$

satisfy assumptions (B1), (B2) and (B3).

Let us begin with the proof of (B1) and (B2). Since the set of tensor product functions

$$\Sigma = \{r \otimes s, r \in V_x, s \in V_t\} = \mathbb{R}\mathcal{D}$$

is assumed to be weakly closed in $(V, \|\cdot\|_V)$ and to satisfy $\overline{\text{Span}(\Sigma)}^{(V, \|\cdot\|_V)} = V$ (assumptions (A1) and (A2)), (B1) and (B2) are direct consequences of the fact that \mathcal{A} and \mathcal{A}^{-1} belong to the space $\mathfrak{L}(V, V)$ (i.e. are bounded operators). For instance, $\mathfrak{L}((V, \|\cdot\|_V), \mathbb{R}) = \mathfrak{L}((V, \|\cdot\|_{\mathcal{A}}), \mathbb{R})$ since

$$\forall l \in \mathfrak{L}((V, \|\cdot\|_V), \mathbb{R}), \frac{1}{\|\mathcal{A}\|_{\mathfrak{L}(V, V)}} \|l\|_{\mathfrak{L}((V, \|\cdot\|_V), \mathbb{R})} \leq \|l\|_{\mathfrak{L}((V, \|\cdot\|_{\mathcal{A}}), \mathbb{R})} \leq \|\mathcal{A}^{-1}\|_{\mathfrak{L}(V, V)} \|l\|_{\mathfrak{L}((V, \|\cdot\|_V), \mathbb{R})}.$$

Let us now prove (B3). Since the operator \mathcal{A} is invertible, the modulus $\rho_{\mathcal{A}}$ of smoothness of $(V, \|\cdot\|_{\mathcal{A}})$ is equal to the modulus of smoothness ρ of $(V, \|\cdot\|_V)$. Indeed, for all $\beta \in \mathbb{R}$,

$$\begin{aligned} \rho_{\mathcal{A}}(\beta) &= \sup_{v, w \in V, \|v\|_{\mathcal{A}} = \|w\|_{\mathcal{A}} = 1} \left(\frac{1}{2} (\|v + \beta w\|_{\mathcal{A}} + \|v - \beta w\|_{\mathcal{A}}) - 1 \right) \\ &= \sup_{v, w \in V, \|\mathcal{A}v\|_V = \|\mathcal{A}w\|_V = 1} \left(\frac{1}{2} (\|\mathcal{A}v + \beta \mathcal{A}w\|_V + \|\mathcal{A}v - \beta \mathcal{A}w\|_V) - 1 \right) \\ &= \sup_{v, w \in V, \|v\|_V = \|w\|_V = 1} \left(\frac{1}{2} (\|v + \beta w\|_V + \|v - \beta w\|_V) - 1 \right) \\ &= \rho(\beta). \end{aligned}$$

Since $(V, \|\cdot\|_V)$ is a Hilbert space,

$$\frac{\rho_{\mathcal{A}}(\beta)}{\beta} = \frac{\rho(\beta)}{\beta} \xrightarrow{\beta \rightarrow 0} 0,$$

and $(V, \|\cdot\|_{\mathcal{A}})$ is a uniformly smooth Banach space.

To implement the X-Greedy or Dual Greedy algorithms in practice in this context, one needs to compute the norm $\|\cdot\|_{\mathcal{A}}$ (see (29) and (31)). Since for all $v \in V$, $\|v\|_{\mathcal{A}} = \|\mathcal{A}v\|_V = \|R_V^{-1} \mathcal{A}v\|_V$, this requires the resolution of several intermediate low- or high-dimensional problems to compute the inverse of the

operator R_V . The same issues as those described in Section 3.2 have to be faced.

Another more practical Banach space

The idea of Lozinski is to replace this norm by a weaker one, easier to compute,

$$\forall v \in V, \|v\|_{i\mathcal{A}} = \sup_{(r,s) \in V_x \times V_t, r \otimes s \neq 0} \frac{a(v, r \otimes s)}{\|r \otimes s\|_V}. \quad (32)$$

Actually, denoting by $\|\cdot\|_i$ the *injective norm* on V [8], defined by

$$\forall v \in V, \|v\|_i = \sup_{(r,s) \in V_x \times V_t, r \otimes s \neq 0} \frac{\langle v, r \otimes s \rangle_V}{\|r \otimes s\|_V},$$

it holds that for all $v \in V$, $\|v\|_{i\mathcal{A}} = \|\mathcal{A}v\|_i$. Reasoning as above, the Banach space $(V, \|\cdot\|_{i\mathcal{A}})$ has exactly the same properties as $(V, \|\cdot\|_i)$.

Since Σ is weakly closed in V , and since for all $v \in V$, $\|v\|_i \leq \|v\|_V$, Σ is also weakly closed in $(V, \|\cdot\|_i)$. But, in the full general case, the Banach space $(V, \|\cdot\|_i)$ and hence the Banach space $(V, \|\cdot\|_{i\mathcal{A}})$ are not uniformly smooth. Actually, these spaces may not even be reflexive. Indeed, let us assume that $V = V_x \otimes V_t$ and that $\|\cdot\|_V$ is the associated cross-norm, in other words that for all $(r, s) \in V_x \times V_t$, $\|r \otimes s\|_V = \|r\|_{V_x} \|s\|_{V_t}$. It holds that [8] $(V, \|\cdot\|_i)$ is isomorphous to $\mathcal{K}(V_x, V_t)$, the Banach space of compact operators from V_x to V_t endowed with the operator norm. Since $\mathcal{K}(V_x, V_t)$ is not a reflexive space ($\mathcal{K}(V_x, V_t)^* = \mathfrak{S}_1(V_x, V_t)$ and $\mathfrak{S}_1(V_x, V_t)^* = \mathfrak{L}(V_x, V_t)$ where $\mathfrak{S}_1(V_x, V_t)$ denotes the set of trace-class operators from V_x to V_t), there is no guarantee of convergence of the relaxed versions of the X-Greedy or Dual Greedy algorithms presented above.

The finite-dimensional cross-norm case

However, in the case when V_x and V_t are finite-dimensional and $V = V_x \otimes V_t$, the spaces $\mathcal{K}(V_x, V_t)$ and $\mathfrak{L}(V_x, V_t)$ are identical. The space $(V, \|\cdot\|_i)$ is then reflexive and uniformly smooth. Indeed, if $V_x = \mathbb{R}^{m_x}$ and $V_t = \mathbb{R}^{m_t}$, since $\|\cdot\|_{V_x}$ and $\|\cdot\|_{V_t}$ both derive from the scalar products $\langle \cdot, \cdot \rangle_{V_x}$ and $\langle \cdot, \cdot \rangle_{V_t}$, there exist two invertible matrices $P \in \mathbb{R}^{m_x \times m_x}$ and $Q \in \mathbb{R}^{m_t \times m_t}$ such that

$$\begin{aligned} \forall r \in \mathbb{R}^{m_x}, \|r\|_{V_x} &= \|Pr\|_{F_{m_x}}, \\ \forall s \in \mathbb{R}^{m_t}, \|s\|_{V_t} &= \|Qs\|_{F_{m_t}}, \end{aligned}$$

where $\|\cdot\|_{F_{m_x}}$ and $\|\cdot\|_{F_{m_t}}$ denote respectively the Frobenius norms on \mathbb{R}^{m_x} and \mathbb{R}^{m_t} . Thus, $(V, \|\cdot\|_i)$ is isometrically isomorphic to $\mathbb{R}^{m_x \times m_t}$ seen as $\mathcal{K}(V_x, V_t)$, endowed with the norm

$$\forall M \in \mathbb{R}^{m_x \times m_t}, \|M\|_i = \sup_{r \in \mathbb{R}^{m_x}, r \neq 0} \frac{\|QMr\|_{F_{m_t}}}{\|Pr\|_{F_{m_x}}} = \|QMP^{-1}\|_2,$$

where

$$\forall M \in \mathbb{R}^{m_x \times m_t}, \|M\|_2 = \sup_{r \in \mathbb{R}^{m_x}, r \neq 0} \frac{\|Mr\|_{F_{m_t}}}{\|r\|_{F_{m_x}}}.$$

Actually, $(\mathbb{R}^{m_x \times m_t}, \|\cdot\|_2)$ is a uniformly smooth Banach space [19]. Thus, greedy algorithms for Banach spaces do converge in this setting.

4.2.3 Practical implementation of the algorithms

X-Greedy algorithm

The X-Greedy algorithm reads:

1. let $u_0 = 0$ and $n = 1$;
2. find $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \|u - u_{n-1} - r \otimes s\|_{i\mathcal{A}}; \quad (33)$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$.

From the definition of the norm $\|\cdot\|_{i\mathcal{A}}$ (see (32)), the second step of the algorithm can be rewritten as

- (2) find $(r_n, s_n) \in V_x \times V_t$ such that

$$\begin{aligned} (r_n, s_n) &\in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \sup_{(\tilde{r}, \tilde{s}) \in V_x \times V_t} \frac{l(\tilde{r} \otimes \tilde{s}) - a(u_{n-1} + r \otimes s, \tilde{r} \otimes \tilde{s})}{\|\tilde{r} \otimes \tilde{s}\|_V} \\ &= \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \sup_{(\tilde{r}, \tilde{s}) \in V_x \times V_t, \|\tilde{r} \otimes \tilde{s}\|_V = 1} l(\tilde{r} \otimes \tilde{s}) - a(u_{n-1} + r \otimes s, \tilde{r} \otimes \tilde{s}). \end{aligned} \quad (34)$$

From a practical point of view, at each iteration $n \in \mathbb{N}^*$, the functions $(r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in V_x^2 \times V_t^2$ are obtained by solving the stationarity equations associated with (34), namely by solving the following coupled problem

$$\begin{cases} \text{find } (r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in V_x^2 \times V_t^2 \text{ such that for all } (\delta r, \delta \tilde{r}, \delta s, \delta \tilde{s}) \in V_x^2 \times V_t^2, \\ \langle \tilde{r}_n \otimes \tilde{s}_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n \rangle_V + a(u_{n-1} + r_n \otimes s_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) = l(\tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n), \\ a(r_n \otimes \delta s + \delta r \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n) = 0. \end{cases}$$

The X-Greedy algorithm has not been implemented in practice yet. Indeed, it is not clear how to compute a solution of the above stationarity equations since using a fixed-point algorithm procedure similar to the one presented in Section 4.1 for the MiniMax algorithm would always lead to $\tilde{r}_n \otimes \tilde{s}_n = 0$, due to the form of the second equation.

Dual Greedy algorithm

Let us describe here how Lozinski adapted the Dual Greedy algorithm for the resolution of high-dimensional non-symmetric linear problems.

A remaining issue concerns the construction of a peak functional $F_{r_{n-1}}$ for the residual $r_{n-1} = u - u_{n-1}$ which is used in the second step of the algorithm (30). Actually, the true peak functional is not computed but only an approximation of this functional by an optimal tensor product function in a sense which is made precise below.

The adapted Dual Greedy algorithm reads:

1. set $u_0 = 0$ and $n = 1$;
2. (computation of an approximate peak functional for the residual $r_{n-1} = u - u_{n-1}$ with a tensor product function) find $(\tilde{r}_n, \tilde{s}_n) \in V_x \times V_t$ such that

$$(\tilde{r}_n, \tilde{s}_n) \in \underset{(\tilde{r}, \tilde{s}) \in V_x \times V_t, \|\tilde{r} \otimes \tilde{s}\|_V = 1}{\operatorname{argmax}} a(u - u_{n-1}, \tilde{r} \otimes \tilde{s}) = \underset{(\tilde{r}, \tilde{s}) \in V_x \times V_t, \|\tilde{r} \otimes \tilde{s}\|_V = 1}{\operatorname{argmax}} l(\tilde{r} \otimes \tilde{s}) - a(u_{n-1}, \tilde{r} \otimes \tilde{s}); \quad (35)$$

3. (Step 2 of the Dual Greedy algorithm, see (30)) find $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r, s) \in V_x \times V_t, \|r \otimes s\|_{i, \mathcal{A}} = 1}{\operatorname{argmax}} a(r \otimes s, \tilde{r}_n \otimes \tilde{s}_n); \quad (36)$$

4. (Step 3 of the Dual Greedy algorithm, see (31)) find $\alpha_n \in \mathbb{R}$ such that

$$\alpha_n \in \operatorname{argmin}_{\alpha \in \mathbb{R}} \|u - u_{n-1} - \alpha r_n \otimes s_n\|_{i\mathcal{A}} = \operatorname{argmin}_{\alpha \in \mathbb{R}} \sup_{(\tilde{r}, \tilde{s}) \in V_x \times V_t, \|\tilde{r} \otimes \tilde{s}\|_V = 1} l(\tilde{r} \otimes \tilde{s}) - a(u_{n-1} + \alpha r_n \otimes s_n, \tilde{r} \otimes \tilde{s}) \quad (37)$$

5. set $u_n = u_{n-1} + \alpha_n r_n \otimes s_n$ and $n = n + 1$.

The Euler-Lagrange equations associated with the minimization problem (35) read: for all $(\delta \tilde{r}, \delta \tilde{s}) \in V_x \times V_t$,

$$\lambda_n \langle \tilde{r}_n \otimes \tilde{s}_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n \rangle_V = l(\tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) - a(u_{n-1}, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n),$$

for some $\lambda_n \in \mathbb{R}$ satisfying $\lambda_n \|\tilde{r}_n \otimes \tilde{s}_n\|_V^2 = \lambda_n = l(\tilde{r}_n \otimes \tilde{s}_n) - a(u_{n-1}, \tilde{r}_n \otimes \tilde{s}_n)$.

The Euler-Lagrange equations associated to the minimization problem (36) can be rewritten as follows: $(r_n, s_n) \in V_x \times V_t$ is solution of

$$a(r_n \otimes \delta s + \delta r \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n) = \mu_n a(r_n \otimes \delta s + \delta r \otimes s_n, \hat{r}_n \otimes \hat{s}_n), \quad \forall (\delta r, \delta s) \in V_x \times V_t,$$

where $(\hat{r}_n, \hat{s}_n) \in V_x \times V_t$ is such that

$$(\hat{r}_n, \hat{s}_n) \in \operatorname{argmax}_{(\hat{r}, \hat{s}) \in V_x \times V_t, \|\hat{r} \otimes \hat{s}\|_V = 1} a(r_n \otimes s_n, \hat{r} \otimes \hat{s}), \quad (38)$$

and $\mu_n = a(r_n \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n)$. Besides, if the pair (\hat{r}_n, \hat{s}_n) satisfies (38), it holds that

$$a(r_n \otimes s_n, \hat{r}_n \otimes \delta \hat{s} + \delta \hat{r} \otimes \hat{s}_n) = \nu_n \langle \hat{r}_n \otimes \hat{s}_n, \hat{r}_n \otimes \delta \hat{s} + \delta \hat{r} \otimes \hat{s}_n \rangle_V, \quad \forall (\delta \hat{r}, \delta \hat{s}) \in V_x \times V_t,$$

with $\nu_n = a(r_n \otimes s_n, \hat{r}_n \otimes \hat{s}_n) = \|r_n \otimes s_n\|_{i\mathcal{A}} = 1$. This yields to a coupled problem on (r_n, s_n) and (\hat{r}_n, \hat{s}_n) . Lozinski the noticed that, if (r_n, s_n) is solution to (36), $(\hat{r}_n, \hat{s}_n) = (\tilde{r}_n, \tilde{s}_n)$ is solution of (38) in the sense that

$$(\tilde{r}_n, \tilde{s}_n) \in \operatorname{argmax}_{(\hat{r}, \hat{s}) \in V_x \times V_t, \|\hat{r} \otimes \hat{s}\|_V = 1} a(r_n \otimes s_n, \hat{r} \otimes \hat{s}).$$

The Euler-Lagrange equations can be rewritten as

$$a(r_n \otimes s_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) = \langle \tilde{r}_n \otimes \tilde{s}_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n \rangle_V, \quad \forall (\delta \tilde{r}, \delta \tilde{s}) \in V_x \times V_t,$$

by taking $\mu_n = a(r_n \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n) = \|r_n \otimes s_n\|_{i\mathcal{A}} = 1$.

The Euler equations associated with (37) read

$$a(u - u_{n-1} - \alpha_n r_n \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n) = 0,$$

yielding to $\alpha_n = a(u - u_{n-1}, \tilde{r}_n \otimes \tilde{s}_n) = \lambda_n$.

Finally, replacing $\alpha_n r_n \otimes s_n$ by $r_n \otimes s_n$, the iterations of the Dual Greedy algorithm are computed in practice as follows:

1. let $u_0 = 0$ and $n = 1$;
2. find $(r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in V_x^2 \times V_t^2$ such that for all $(\delta r, \delta \tilde{r}, \delta s, \delta \tilde{s}) \in V_x^2 \times V_t^2$,
$$\begin{cases} \langle \tilde{r}_n \otimes \tilde{s}_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n \rangle_V = l(\tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) - a(u_{n-1}, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n), \\ a(r_n \otimes s_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) = l(\tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) - a(u_{n-1}, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n); \end{cases}$$
3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$.

These equations lead to two decoupled problems on (r_n, s_n) and $(\tilde{r}_n, \tilde{s}_n)$. Each of them is solved through a fixed-point procedure similar to (15) described in details in Section 6.3.1. Some numerical tests are presented in Section 6, which illustrate the convergence of this algorithm.

5 The Decomposition algorithm

Let us now present a new algorithm based on a *decomposition* of the bilinear form a . The bilinear form a can always be written as

$$a(\cdot, \cdot) = b_s(\cdot, \cdot) + b(\cdot, \cdot) \tag{39}$$

where $b_s(\cdot, \cdot)$ is a symmetric coercive continuous bilinear form on $V \times V$ and $b(\cdot, \cdot)$ a (not necessarily symmetric) continuous bilinear form on $V \times V$. In the sequel, $b_s(\cdot, \cdot)$ (respectively $b(\cdot, \cdot)$) will be referred to as the *implicit* (respectively *explicit*) part of $a(\cdot, \cdot)$. This terminology will be explained below.

Example 5.1. Let us introduce a_s (respectively a_{as}) the symmetric part (respectively antisymmetric part) of $a(\cdot, \cdot)$, defined by:

$$\forall v, w \in V, \quad a_s(v, w) = \frac{1}{2} (a(v, w) + a(w, v)), \tag{40}$$

and

$$\forall v, w \in V, a_{as}(v, w) = \frac{1}{2} (a(v, w) - a(w, v)). \quad (41)$$

Provided that a_s is coercive, the decomposition $a(\cdot, \cdot) = a_s(\cdot, \cdot) + a_{as}(\cdot, \cdot)$ is admissible in the sense of (39).

Let us denote by \mathcal{B}_s and \mathcal{B} the bounded operators on V defined by

$$\begin{aligned} \forall v, w \in V, b_s(v, w) &= \langle \mathcal{B}_s v, w \rangle_V, \\ \forall v, w \in V, b(v, w) &= \langle \mathcal{B} v, w \rangle_V. \end{aligned}$$

Since b_s is coercive, the operator \mathcal{B}_s is invertible.

The principle of the algorithm we consider in this section to solve problem (16) consists in *expliciting* the part b of the bilinear form as a right-hand side source term. More precisely, one can consider the following fixed-point algorithm:

1. choose a starting guess $u_0 \in V$ and set $n = 1$;
2. let u_n be the unique solution to

$$\forall v \in V, b_s(u_n, v) = l(v) - b(u_{n-1}, v); \quad (42)$$

3. set $n = n + 1$.

In other words, $u_n = F(u_{n-1})$ where for all $v \in V$, $F(v) = \mathcal{B}_s^{-1}(\mathcal{L} - \mathcal{B}v)$. If $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathcal{L}(V,V)} < 1$, the mapping $F : V \rightarrow V$ is a contraction and it follows from the Picard fixed-point theorem that the sequence $(u_n)_{n \in \mathbb{N}}$ strongly converges in V towards the solution u of the initial problem (16).

A natural approach thus consists in solving problem (42) at each iteration $n \in \mathbb{N}^*$ using a standard greedy procedure. Provided that the greedy expansion obtained at each iteration $n \in \mathbb{N}^*$ is accurate enough, the sequence $(u_n)_{n \in \mathbb{N}}$ given by this algorithm strongly converges in V towards u .

However, the principle of this method requires to compute a full greedy loop at each iteration $n \in \mathbb{N}^*$. In order to save computational time, we now introduce the following algorithm, in which only one tensor product function is computed at each iteration $n \in \mathbb{N}^*$.

Decomposition algorithm:

1. let $u_0 = 0$ and $n = 1$;
2. find $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \frac{1}{2} b_s(u_{n-1} + r \otimes s, u_{n-1} + r \otimes s) - l(r \otimes s) - b(u_{n-1}, r \otimes s); \quad (43)$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and set $n = n + 1$.

The bilinear form $b(\cdot, \cdot)$ is *explicited* as a right-hand side, whereas $b_s(\cdot, \cdot)$ remains *implicit*. This justifies the terminology introduced in the beginning of the section.

Equation (43) can be rewritten equivalently as

$$\begin{aligned} (r_n, s_n) &\in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \frac{1}{2} b_s(r \otimes s, r \otimes s) - l(r \otimes s) - b_s(u_{n-1} + r \otimes s) - b(u_{n-1}, r \otimes s) \\ &= \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \frac{1}{2} b_s(r \otimes s, r \otimes s) - l(r \otimes s) - a(u_{n-1}, r \otimes s). \end{aligned}$$

This means that for each $n \in \mathbb{N}^*$, $r_n \otimes s_n$ is a tensor product solution to the first iteration of the greedy algorithm applied to the following symmetric coercive problem

$$\begin{cases} \text{find } u \in V \text{ such that} \\ \forall v \in V, \quad b_s(u, v) = l(v) - b_s(u_{n-1}, v) - b(u_{n-1}, v) = l(v) - a(u_{n-1}, v). \end{cases}$$

As explained above, such an algorithm is expected to converge only if the norm $\|\mathcal{B}_s^{-1} \mathcal{B}\|_{\mathfrak{L}(V,V)}$ is small enough. Actually, in the case when the spaces V_x and V_t are finite dimensional, the following result holds:

Proposition 5.1. *If V_x and V_t are finite dimensional, there exists $\kappa > 0$ such that if*

$$\|\mathcal{B}_s^{-1} \mathcal{B}\|_{\mathfrak{L}(V,V)} \leq \kappa, \quad (44)$$

then the sequence $(u_n)_{n \in \mathbb{N}^}$ defined by the Decomposition algorithm strongly converges to u in V .*

Proof. Since V_t and V_x are assumed to be finite dimensional, from assumption (A1), so is V . Let $\kappa := \|\mathcal{B}_s^{-1} \mathcal{B}\|_{\mathfrak{L}(V,V)}$. Let us denote by $\langle \cdot, \cdot \rangle_{\tilde{V}} = b_s(\cdot, \cdot)$ the scalar product on V induced by the symmetric bilinear form b_s , and by $\|\cdot\|_{\tilde{V}}$ the associated norm.

Let $\tilde{\mathcal{L}} \in V$ and $\tilde{\mathcal{B}} \in \mathfrak{L}(V, V)$ be defined by

$$\begin{aligned} \forall v \in V, \quad l(v) &= \langle \tilde{\mathcal{L}}, v \rangle_{\tilde{V}}, \\ \forall v, w \in V, \quad b(v, w) &= \langle \tilde{\mathcal{B}}v, w \rangle_{\tilde{V}}. \end{aligned}$$

Actually, $\tilde{\mathcal{B}} = \mathcal{B}_s^{-1}\mathcal{B}$ and $\tilde{\mathcal{L}} = \mathcal{B}_s^{-1}\mathcal{L}$. This implies that for all $v \in V$, $\|\tilde{\mathcal{A}}v\|_{\tilde{V}} \leq \kappa\|v\|_{\tilde{V}}$. Besides, $\mathcal{I} + \tilde{\mathcal{B}} = \mathcal{B}_s^{-1}(\mathcal{B}_s + \mathcal{B}) = \mathcal{B}_s^{-1}\mathcal{A}$, where \mathcal{I} denotes the identity operator on V .

The proof relies on the fact that all norms are equivalent in finite dimension. In particular, let us define the injective norm $\|\cdot\|_{\tilde{\mathcal{I}}}$ by

$$\forall v \in V, \quad \|v\|_{\tilde{\mathcal{I}}} = \sup_{(r,s) \in V_x \times V_t} \frac{\langle v, r \otimes s \rangle_{\tilde{V}}}{\|r \otimes s\|_{\tilde{V}}}.$$

Then, there exists $\alpha > 0$ such that

$$\forall v \in V, \quad \|v\|_{\tilde{V}} \leq \alpha\|v\|_{\tilde{\mathcal{I}}}.$$

Besides, for all $v \in V$, $\|v\|_{\tilde{\mathcal{I}}} \leq \|v\|_{\tilde{V}}$.

Let us introduce $\tilde{\mathcal{U}}_n := \tilde{\mathcal{L}} - \tilde{\mathcal{B}}u_n - u_n = \mathcal{B}_s^{-1}(\mathcal{L} - \mathcal{A}u_n)$. For all $n \in \mathbb{N}^*$, $\tilde{\mathcal{U}}_n$ is the vector of V such that

$$\forall v \in V, \quad \langle \tilde{\mathcal{U}}_n, v \rangle_{\tilde{V}} = l(v) - a(u_n, v).$$

For $n \geq 1$, the Euler equations associated to (43) read:

$$b_s(u_{n-1} + r_n \otimes s_n, r_n \otimes \delta s + \delta r \otimes s_n) = l(r_n \otimes \delta s + \delta r \otimes s_n) - b(u_{n-1}, r_n \otimes \delta s + \delta r \otimes s_n), \quad \forall (\delta r, \delta s) \in V_x \times V_t.$$

As a consequence, $l(r_n \otimes s_n) - a(u_{n-1}, r_n \otimes s_n) - b_s(r_n \otimes s_n) = \langle \tilde{\mathcal{U}}_{n-1} - r_n \otimes s_n, r_n \otimes s_n \rangle_{\tilde{V}} = 0$, which implies that $\|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{V}}^2 = \|\tilde{\mathcal{U}}_{n-1} - r_n \otimes s_n\|_{\tilde{V}}^2 + \|r_n \otimes s_n\|_{\tilde{V}}^2$. Furthermore, from Lemma 1.1, we have

$$\|r_n \otimes s_n\|_{\tilde{V}} = \sup_{(r,s) \in V_x \times V_t, r \otimes s \neq 0} \frac{\langle \tilde{\mathcal{U}}_{n-1}, r \otimes s \rangle_{\tilde{V}}}{\|r \otimes s\|_{\tilde{V}}} = \|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{\mathcal{I}}}.$$

It holds that

$$\begin{aligned}
\|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{V}}^2 - \|\tilde{\mathcal{U}}_n\|_{\tilde{V}}^2 &= \|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{V}}^2 - \|\tilde{\mathcal{U}}_{n-1} - r_n \otimes s_n - \tilde{\mathcal{B}}r_n \otimes s_n\|_{\tilde{V}}^2 \\
&= \|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{V}}^2 - \|\tilde{\mathcal{U}}_{n-1} - r_n \otimes s_n\|_{\tilde{V}}^2 - \|\tilde{\mathcal{B}}r_n \otimes s_n\|_{\tilde{V}}^2 \\
&\quad + 2\langle \tilde{\mathcal{B}}r_n \otimes s_n, \tilde{\mathcal{U}}_{n-1} - r_n \otimes s_n \rangle_{\tilde{V}} \\
&= \|r_n \otimes s_n\|_{\tilde{V}}^2 - \|\tilde{\mathcal{B}}r_n \otimes s_n\|_{\tilde{V}}^2 - 2\langle \tilde{\mathcal{B}}r_n \otimes s_n, \tilde{\mathcal{U}}_{n-1} - r_n \otimes s_n \rangle_{\tilde{V}} \\
&\geq (1 - \kappa^2)\|r_n \otimes s_n\|_{\tilde{V}}^2 - 2\kappa\|r_n \otimes s_n\|_{\tilde{V}}\|\tilde{\mathcal{U}}_{n-1} - r_n \otimes s_n\|_{\tilde{V}} \\
&\geq (1 - \kappa^2)\|r_n \otimes s_n\|_{\tilde{V}}^2 - 2\kappa\|r_n \otimes s_n\|_{\tilde{V}}\|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{V}} \\
&\geq (1 - \kappa^2)\|r_n \otimes s_n\|_{\tilde{V}}^2 - 2\kappa\alpha\|r_n \otimes s_n\|_{\tilde{V}}\|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{t}} \\
&= (1 - \kappa^2 - 2\alpha\kappa)\|r_n \otimes s_n\|_{\tilde{V}}^2 \\
&= (1 - \kappa^2 - 2\alpha\kappa)\|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{t}}^2.
\end{aligned}$$

If κ is small enough to ensure that $1 - \kappa^2 - 2\alpha\kappa > 0$, the sequence $(\|\tilde{\mathcal{U}}_n\|_{\tilde{V}})_{n \in \mathbb{N}^*}$ is non-increasing, hence convergent. Thus, the series of general terms $(\|\tilde{\mathcal{U}}_{n-1}\|_{\tilde{V}}^2 - \|\tilde{\mathcal{U}}_n\|_{\tilde{V}}^2)_{n \in \mathbb{N}^*}$ and $(\|\tilde{\mathcal{U}}_n\|_{\tilde{t}}^2)_{n \in \mathbb{N}^*}$ are convergent. This yields

$$\|\tilde{\mathcal{U}}_n\|_{\tilde{t}} \xrightarrow{n \rightarrow \infty} 0,$$

and, as all norms are equivalent in finite dimension,

$$\|\tilde{\mathcal{U}}_n\|_{\tilde{V}} \xrightarrow{n \rightarrow \infty} 0.$$

If $\kappa < 1$, the operator $(\mathcal{I} + \tilde{\mathcal{B}})^{-1}$ is continuous from V to V for the norm $\|\cdot\|_{\tilde{V}}$, and

$$\|u - u_n\|_{\tilde{V}} = \|\mathcal{A}^{-1}\mathcal{L} - u_n\|_{\tilde{V}} = \left\| (\mathcal{B}_s^{-1}\mathcal{A})^{-1} \mathcal{B}_s^{-1}\mathcal{L} - u_n \right\|_{\tilde{V}} = \|(\mathcal{I} + \tilde{\mathcal{B}})^{-1}(\tilde{\mathcal{L}} - (\mathcal{I} + \tilde{\mathcal{B}})u_n)\|_{\tilde{V}} = \|(\mathcal{I} + \tilde{\mathcal{B}})^{-1}\tilde{\mathcal{L}} - u_n\|_{\tilde{V}}.$$

Since the norm $\|\cdot\|_{\tilde{V}}$ is equivalent to the norm $\|\cdot\|_V$, we obtain the desired result. \square

Unfortunately, the rate κ in (44) strongly depends on the dimensions of V_x and V_t , as shown in the proof of Proposition 5.1. However, in some simple numerical experiments we performed with this algorithm, the rate κ does not

seem to depend on the dimension, as illustrated in Section 6. The analysis of this numerical observation is work in progress.

A possible way to obtain a decomposition such as (39) satisfying (44) is to consider preconditioners for the initial antisymmetric problem. Problem (16) can be rewritten as

$$\begin{cases} \text{find } u \in V \text{ such that} \\ u = (\mathcal{I} - \mathcal{CA})u + \mathcal{CL}, \end{cases} \quad (45)$$

where \mathcal{C} is a well-chosen continuous linear operator on V such that $\|\mathcal{I} - \mathcal{CA}\|_{\mathcal{L}(V,V)}$ is as small as possible. With this formulation, one can consider the following algorithm:

1. let $u_0 = 0$ and $n = 1$;
2. find $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \frac{1}{2} \langle u_{n-1} + r \otimes s, u_{n-1} + r \otimes s \rangle_V - \langle (\mathcal{I} - \mathcal{CA})u_{n-1} + \mathcal{CL}, r \otimes s \rangle_V; \quad (46)$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$.

The bilinear form associated with the continuous operator \mathcal{CA} needs to be easy to compute on tensor product functions. At least in the case when V is finite dimensional, if $\|\mathcal{I} - \mathcal{CA}\|_{\mathcal{L}(V,V)}$ is small enough, the sequence $(u_n)_{n \in \mathbb{N}^*}$ converges strongly in V towards the solution u of (16). However, finding a suitable operator \mathcal{C} is not an easy task in general.

6 Numerical results

6.1 Presentation of the toy problems

In this section, we present the two toy problems we consider in these numerical tests. Let $b_x, b_t \in \mathbb{R}$, $m_x = m_t = 1$ and $\mathcal{X} = \mathcal{T} = (-1, 1)$.

The first toy problem is inspired from Example 2.1. In the continuous setting, we define

- $V := L^2_{\text{per}}((-1, 1), H^1_{\text{per}}((-1, 1), \mathbb{C})) = L^2_{\text{per}}((-1, 1), \mathbb{C}) \otimes H^1_{\text{per}}((-1, 1), \mathbb{C})$;

- $V_x := H_{\text{per}}^1((-1, 1), \mathbb{C});$
- $V_t := L_{\text{per}}^2((-1, 1), \mathbb{C});$
- $f \in L_{\text{per}}^2((-1, 1)^2, \mathbb{C});$
- for all $u, v \in V,$

$$a(u, v) := \int_{-1}^1 \int_{-1}^1 (\nabla_x u \cdot \overline{\nabla_x v} + (b_x \cdot \nabla_x u) \overline{v} + u \overline{v}),$$

and

$$l(v) := \int_{-1}^1 \int_{-1}^1 f \overline{v}.$$

The second toy problem is the following

- $V := H_{\text{per}}^1((-1, 1)^2, \mathbb{C});$
- $V_x := H_{\text{per}}^1((-1, 1), \mathbb{C});$
- $V_t := H_{\text{per}}^1((-1, 1), \mathbb{C});$
- $f \in L_{\text{per}}^2((-1, 1)^2, \mathbb{C});$
- for all $u, v \in V,$

$$\begin{aligned} a(u, v) &:= \int_{-1}^1 \int_{-1}^1 (\nabla_x u \cdot \overline{\nabla_x v} + \nabla_t u \cdot \overline{\nabla_t v}) \\ &\quad + \int_{-1}^1 \int_{-1}^1 ((b_x \cdot \nabla_x u + b_t \cdot \nabla_t u) \overline{v} + u \overline{v}), \end{aligned}$$

and

$$l(v) := \int_{-1}^1 \int_{-1}^1 f \overline{v}.$$

It can be easily checked in the two cases that the Hilbert spaces V , V_x and V_t satisfy assumptions (A1) and (A2). Besides, the sesquilinear form $a : V \times V \rightarrow \mathbb{C}$ is continuous and satisfies assumption (A3). Our aim is to approximate the function $u \in V$ such that

$$\forall v \in V, a(u, v) = l(v). \quad (47)$$

We introduce finite-dimensional discretization spaces defined as follows. For all $k \in \mathbb{Z}$, let $e_k^x : \mathcal{X} \ni x \mapsto \frac{1}{\sqrt{2}}e^{ikx}$ and $e_k^t : \mathcal{T} \ni t \mapsto \frac{1}{\sqrt{2}}e^{ikt}$. For $N_x, N_t \in \mathbb{N}^*$, we define

$$V_x^{N_x} := \text{Span} \{e_k^x, -N_x \leq k \leq N_x\},$$

$$V_t^{N_t} := \text{Span} \{e_k^t, -N_t \leq k \leq N_t\},$$

and $V^{N_x, N_t} := V_x^{N_x} \otimes V_t^{N_t}$.

We then consider the Galerkin approximation of problem (47) in the discretization space V^{N_x, N_t} , i.e. find $u \in V^{N_x, N_t}$ such that

$$\forall v \in V^{N_x, N_t}, \quad a(u, v) = l(v). \quad (48)$$

For u solution of (48), we denote by $U = (u_{kl})_{|k| \leq N_x, |l| \leq N_t} \in \mathbb{C}^{(2N_x+1) \times (2N_t+1)}$ the matrix such that

$$u = \sum_{k=-N_x}^{N_x} \sum_{l=-N_t}^{N_t} u_{kl} e_k^x \otimes e_l^t,$$

and by $F = (f_{kl})_{|k| \leq N_x, |l| \leq N_t} \in \mathbb{C}^{(2N_x+1) \times (2N_t+1)}$ the matrix such that

$$f = \sum_{k=-N_x}^{N_x} \sum_{l=-N_t}^{N_t} f_{kl} e_k^x \otimes e_l^t.$$

In this discrete setting, the first problem is equivalent to: find $U \in \mathbb{C}^{(2N_x+1) \times (2N_t+1)}$ such that

$$D_x U + b_x N_x U + U = F$$

and the second problem equivalent to: find $U \in \mathbb{C}^{(2N_x+1) \times (2N_t+1)}$ such that

$$D_x U + b_x N_x U + U D_t + b_t U N_t + U = F \quad (49)$$

where $D_x, N_x \in \mathbb{C}^{(2N_x+1) \times (2N_x+1)}$, $D_t, N_t \in \mathbb{C}^{(2N_t+1) \times (2N_t+1)}$, and for all $-N_x \leq k, k' \leq N_x$ and $-N_t \leq l, l' \leq N_t$,

$$\begin{aligned} (D_x)_{kk'} &= |k|^2 \delta_{kk'}, \\ (D_t)_{ll'} &= |l|^2 \delta_{ll'}, \\ (N_x)_{kk'} &= ik \delta_{kk'}, \\ (N_t)_{ll'} &= il \delta_{ll'}. \end{aligned}$$

6.2 Tests with the Decomposition algorithm

Let us begin with the numerical tests performed to illustrate the convergence of the Decomposition algorithm presented in Section 5.

6.2.1 The fixed-point loop

Before presenting the numerical results obtained with this algorithm, we would like to discuss the fixed-point procedure used in practice in order to compute the pair of functions $(r_n, s_n) \in V_t^{N_t} \times V_x^{N_x}$, solution of (43) at each iteration $n \in \mathbb{N}^*$ of the Decomposition algorithm.

The algorithm reads as follows:

- choose $(r_n^{(0)}, s_n^{(0)}) \in V_t^{N_t} \times V_x^{N_x}$ and set $m = 1$;
- find $(r_n^{(m)}, s_n^{(m)}) \in V_t^{N_t} \times V_x^{N_x}$ such that

$$\begin{cases} b_s \left(r_n^{(m)} \otimes s_n^{(m-1)}, \delta r \otimes s_n^{(m-1)} \right) = l \left(\delta r \otimes s_n^{(m-1)} \right) - a \left(u_{n-1}, \delta r \otimes s_n^{(m-1)} \right), \forall \delta r \in V_x^{N_x}, \\ b_s \left(r_n^{(m)} \otimes s_n^{(m)}, r_n^{(m)} \otimes \delta s \right) = l \left(r_n^{(m)} \otimes \delta s \right) - a \left(u_{n-1}, r_n^{(m)} \otimes \delta s \right), \forall \delta s \in V_t^{N_t}; \end{cases}$$

- set $m = m + 1$.

This fixed-point procedure exhibits of the exponential convergence rate which is numerically observed for standard greedy algorithms for symmetric coercive problems, as discussed in Section 1. The stopping criterion we choose for simulations presented in Section 6.2.2 is the following: $\|r_n^{(m)} \otimes s_n^{(m)} - r_n^{(m-1)} \otimes s_n^{(m-1)}\| < \varepsilon$ where $\|\cdot\|$ denotes the Frobenius norm and $\varepsilon = 10^{-8}$.

6.2.2 Numerical results

We consider the second problem presented in Section 6.1, where $f \in L_{\text{per}}^2((-1, 1)^2, \mathbb{C})$ is chosen such that

$$f = \sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} f_{kl} e_k^x \otimes e_l^t,$$

with $f_{kl} = \frac{1}{|k|^2 + |l|^2 + 1}$ for all $k, l \in \mathbb{Z}$.

We decompose the bilinear form $a(\cdot, \cdot)$ as $a(\cdot, \cdot) = b_s(\cdot, \cdot) + b(\cdot, \cdot)$ where $b_s(\cdot, \cdot) = a_s(\cdot, \cdot)$ is the symmetric part of $a(\cdot, \cdot)$ defined in (40) and $b(\cdot, \cdot) = a_{as}(\cdot, \cdot)$ is the antisymmetric part of $a(\cdot, \cdot)$ defined in (41).

Let us recall that Proposition 5.1 states that, in the finite-dimensional case, there exists a rate κ (which depends on the dimension of the Hilbert spaces) small enough such that if $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathfrak{L}(V,V)} < \kappa$, the Decomposition algorithm is ensured to converge. In the numerical simulations presented below, we have witnessed that there exists a threshold rate κ such that

- if $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathfrak{L}(V,V)} < \kappa$, the algorithm converges;
- if $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathfrak{L}(V,V)} = \kappa$, the algorithm does not converge, but the norm of the residual remains bounded;
- if $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathfrak{L}(V,V)} > \kappa$, the algorithm does not converge, and the norm of the residual blows up.

Besides, the rate κ seems to be independent of the dimension of the Hilbert spaces.

For $n \in \mathbb{N}^*$, let $U_n \in \mathbb{C}^{(2N_x+1) \times (2N_t+1)}$ denote the approximation of U solution of (49) given by the algorithm at the n^{th} iteration. The following three figures show the evolution of the logarithm of the norm of the residual $\|F - D_x U_n - U_n D_t - b_x N_x U_n - b_t U_n N_t - U\|_{\mathfrak{S}_2}$ (where $\|\cdot\|_{\mathfrak{S}_2}$ denotes the Frobenius norm), for different values of $N = N_x = N_t$ and $b = b_x = b_t$ as a function of n . We observe numerically that in this case, the limiting rate is obtained for $b = 1.5$ for any value of N .

We also performed another set of tests with a modified version of the Decomposition algorithm, in order to *increase the threshold rate κ* , in a sense which will be precised below. The modified Decomposition algorithm reads as follows for $\alpha \in [1, +\infty)$:

1. let $u_0 = 0$ and $n = 1$;
2. find $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \frac{\alpha}{2} b_s(r \otimes s, r \otimes s) - l(r \otimes s) - a(u_{n-1}, r \otimes s); \quad (50)$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and set $n = n + 1$.

Let us point out that this algorithms is equivalent to the standard Decomposition algorithm presented in Section 5 in the case when $\alpha = 1$.

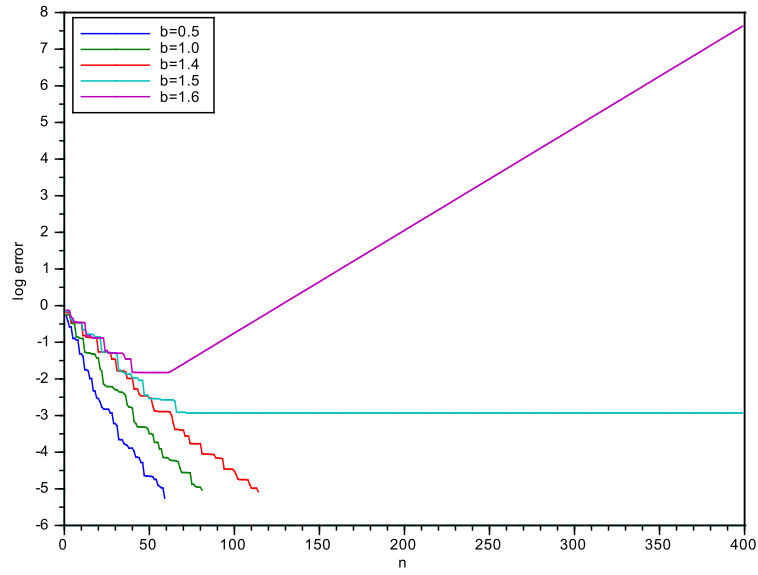


Figure 2: Evolution of $\log_{10} (\|F - D_x U_n - U_n D_t - b_x N_x U_n - b_t U_n N_t - U\|_{\mathfrak{S}_2})$ as a function of n for $N = 20$ and different values of b .

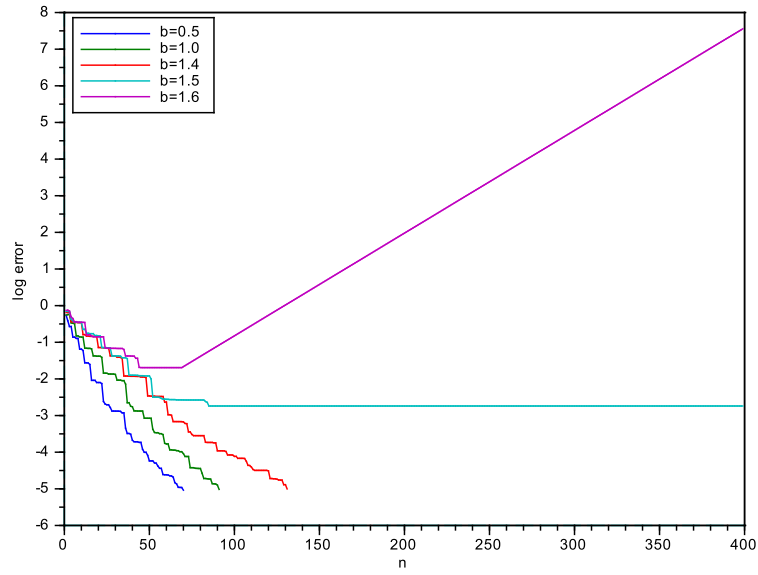


Figure 3: Evolution of $\log_{10} (\|F - D_x U_n - U_n D_t - b_x N_x U_n - b_t U_n N_t - U\|_{\mathfrak{S}_2})$ as a function of n for $N = 50$ and different values of b .

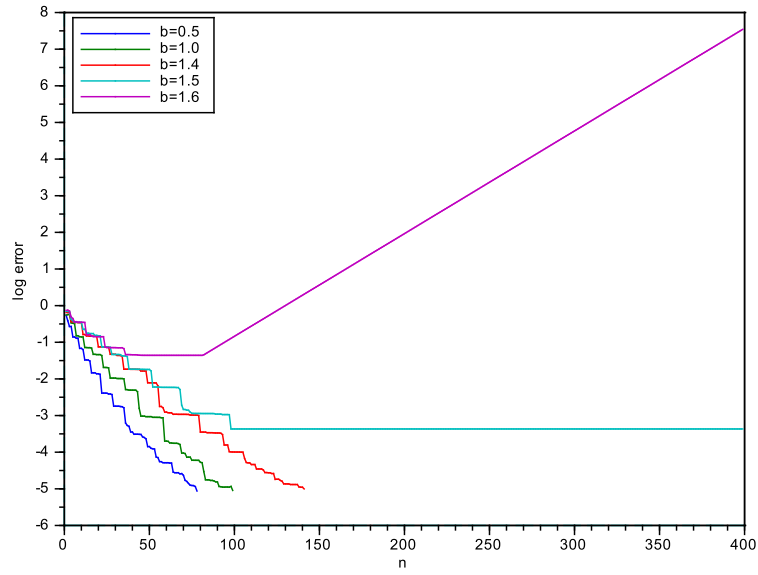


Figure 4: Evolution of $\log_{10} (\|F - D_x U_n - U_n D_t - b_x N_x U_n - b_t U_n N_t - U\|_{\mathfrak{S}_2})$ as a function of n for $N = 100$ and different values of b .

Equivalently, for each $n \in \mathbb{N}^*$, $r_n \otimes s_n$ is a tensor product solution to the first iteration of the greedy algorithm applied to the symmetric coercive problem

$$\begin{cases} \text{find } u \in V \text{ such that} \\ \forall v \in V, \alpha b_s(u, v) = l(v) - a(u_{n-1}, v). \end{cases}$$

We observe that this algorithm has the same convergence properties as the standard Decomposition algorithm, i.e. there exists a threshold rate $\kappa_\alpha = \|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathcal{L}(V)}$ such that

- if $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathcal{L}(V,V)} < \kappa_\alpha$, the algorithm converges;
- if $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathcal{L}(V,V)} = \kappa_\alpha$, the algorithm does not converge, but the norm of the residual remains bounded;
- if $\|\mathcal{B}_s^{-1}\mathcal{B}\|_{\mathcal{L}(V,V)} > \kappa_\alpha$, the algorithm does not converge, and the norm of the residual blows up.

The rate κ_α also seems not to depend on the dimension of the Hilbert space. Besides, $\alpha \in [1, +\infty) \mapsto \kappa_\alpha$ seems to be an increasing function. Thus, choosing a larger value of α seems to lead to an algorithm which is convergent for larger values of b . However, the larger α , the smaller the rate of convergence of the algorithm for a given value of N and b .

The figure below presents the evolution of the logarithm of the norm of the residual $\|F - D_x U_n - U_n D_t - b_x N_x U_n - b_t U_n N_t - U\|_{\mathfrak{S}_2}$ for the second problem for $\alpha = 2$, $N = 50$ and different values of b . The threshold value of b seems to be in this case $b = 2.6$.

6.3 Comparison of Galerkin, Minimax, Dual Greedy and Decomposition algorithms

In this section, we present various numerical tests performed to compare the performances of the Galerkin, Minimax, Dual Greedy and Decomposition algorithms.

6.3.1 Fixed point procedures

Before presenting the simulations done to compare the performance of the four algorithms, we will detail more precisely the fixed-point procedures used for the Galerkin, Dual and MiniMax algorithms.

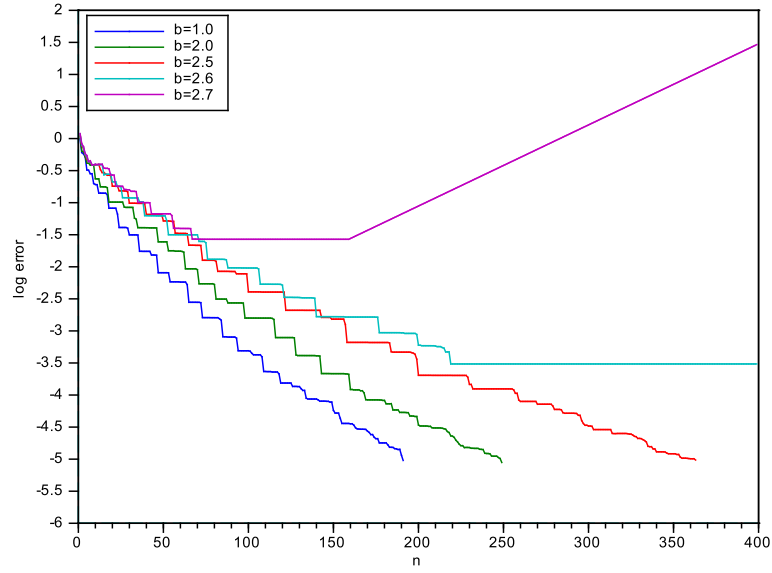


Figure 5: Evolution of $\log_{10} (\|F - D_x U_n - U_n D_t - b_x N_x U_n - b_t U_n N_t - U\|_{\mathfrak{S}_2})$ as a function of n for the modified Decomposition algorithm with $\alpha = 2$, $N = 50$ and different values of b .

Let us first present the fixed-point procedure used in the Galerkin algorithm. For all $n \in \mathbb{N}^*$, the method used to compute a pair $(r_n, s_n) \in V_x \times V_t$ solution to (19) is the following:

- choose $(r_n^{(0)}, s_n^{(0)}) \in V_x \times V_t$ and set $m = 1$;

- find $(r_n^{(m)}, s_n^{(m)}) \in V_x \times V_t$ such that

$$\begin{cases} a(r_n^{(m)} \otimes s_n^{(m-1)}, \delta r \otimes s_n^{(m-1)}) = l(\delta r \otimes s_n^{(m-1)}) - a(u_{n-1}, \delta r \otimes s_n^{(m-1)}), \forall \delta r \in V_x^{N_x}, \\ a(r_n^{(m)} \otimes s_n^{(m)}, r_n^{(m)} \otimes \delta s) = l(r_n^{(m)} \otimes \delta s) - a(u_{n-1}, r_n^{(m)} \otimes \delta s), \forall \delta s \in V_t^{N_t}; \end{cases}$$

- set $m = m + 1$.

All the iterations of the fixed-point procedure are well-defined. However, we have seen from Example 2.1 that there are cases where any solution $(r_n \otimes s_n) \in V_x \times V_t$ of

$$\forall (\delta r, \delta s) \in V_x \times V_t, a(r_n \otimes s_n, r_n \otimes \delta s + \delta r \otimes s_n) = l(r_n \otimes \delta s + \delta r \otimes s_n) - a(u_{n-1}, r_n \otimes \delta s + \delta r \otimes s_n),$$

satisfies $r_n \otimes s_n = 0$. From Example 2.1, this is the case in particular for the first problem presented in Section 6.1 with $f \in L_{\text{per}}^2((-1, 1)^2, \mathbb{C})$ being chosen such that for almost all $x, t \in (-1, 1)$, $f(x, t) = \phi(x - t)$ with $\phi \in L_{\text{per}}^2((-1, 1), \mathbb{C})$ a real-valued odd function. The function $f = \sum_{k, l \in \mathbb{Z}} f_{kl} e_k^x \otimes e_l^t$ with $f_{kl} = \delta_{1,k} \delta_{-1,l} + \delta_{-1,k} \delta_{1,l}$ for all $k, l \in \mathbb{Z}$ is an example of such a function. On this particular example, we observe numerically that, for $n = 1$, the fixed-point procedure presented above does not converge.

The practical implementation of the Galerkin algorithm requires the use of another stopping criterion as the one used in Section 6.2.1. In the numerical simulations presented in Section 6.3.2, and in the fixed-point procedures used in all the other algorithms implemented, the stopping criterion will be $m < m_{\max}$ with $m_{\max} = 20$.

The fixed-point algorithm used for the Decomposition algorithm has been detailed in Section 6.2.1, and the one used for the MiniMax algorithm has been described in Section 4.1. The fixed-point procedure for the Dual Greedy algorithm reads as follows:

- choose $(\tilde{r}_n^{(0)}, \tilde{s}_n^{(0)}) \in V_x \times V_t$ and set $\tilde{m} = 1$;

- while $\tilde{m} < m_{max}$, find $(\tilde{r}_n^{(m)}, \tilde{s}_n^{(m)}) \in V_x \times V_t$ such that for all $(\delta\tilde{r}, \delta\tilde{s}) \in V_x \times V_t$,

$$\begin{cases} \langle \tilde{r}_n^{(m)} \otimes \tilde{s}_n^{(m-1)}, \delta\tilde{r} \otimes \tilde{s}_n^{(m-1)} \rangle_V = l(\delta\tilde{r} \otimes \tilde{s}_n^{(m-1)}) - a(u_{n-1}, \delta\tilde{r} \otimes \tilde{s}_n^{(m-1)}), \\ \langle \tilde{r}_n^{(m)} \otimes \tilde{s}_n^{(m)}, \tilde{r}_n^{(m)} \otimes \delta\tilde{s} \rangle_V = l(\tilde{r}_n^{(m)} \otimes \delta\tilde{s}) - a(u_{n-1}, \tilde{r}_n^{(m)} \otimes \delta\tilde{s}); \end{cases}$$
- set $\tilde{m} = \tilde{m} + 1$;
- if $\tilde{m} \geq m_{max}$, set $\tilde{r}_n = \tilde{r}_n^{(\tilde{m})}$ and $\tilde{s}_n = \tilde{s}_n^{(\tilde{m})}$;
- choose $(r_n^{(0)}, s_n^{(0)}) \in V_x \times V_t$ and set $m = 1$;
- while $m < m_{max}$, find $(r_n^{(m)}, s_n^{(m)}) \in V_x \times V_t$ such that for all $(\delta\tilde{r}, \delta\tilde{s}) \in V_x \times V_t$,

$$\begin{cases} a(r_n^{(m)} \otimes s_n^{(m-1)}, \delta\tilde{r} \otimes \tilde{s}_n) = l(\delta\tilde{r} \otimes \tilde{s}_n) - a(u_{n-1}, \delta\tilde{r} \otimes \tilde{s}_n), \\ a(r_n^{(m)} \otimes s_n^{(m)}, \tilde{r}_n \otimes \delta\tilde{s}) = l(\tilde{r}_n \otimes \delta\tilde{s}) - a(u_{n-1}, \tilde{r}_n \otimes \delta\tilde{s}); \end{cases}$$
- set $m = m + 1$;
- if $m \geq m_{max}$, set $r_n = r_n^{(m)}$ and $s_n = s_n^{(m)}$.

6.3.2 Numerical results

We present here some numerical results obtained for the second problem introduced in Section 6.1, with f chosen as in Section 6.2.2. Here $N = N_t = N_x = 50$ and the different figures show the rates of convergence of the different algorithms for several values of $b = b_x = b_t$. The different values of b are the following: 0.01, 0.1, 1 and 2. When $b = 2$, the Decomposition algorithm does not converge.

We observe numerically that even if the Galerkin algorithm is not well-defined through the use of the equations (19), using a fixed-point procedure as described above and stopping the procedure after a number m_{max} of iterations is enough to observe good convergence properties of the algorithm. Besides, the Dual Greedy and MiniMax algorithm are almost as efficient as the Galerkin algorithm.

The Decomposition algorithm is very efficient when the antisymmetric part of the bilinear form $a(\cdot, \cdot)$ is small, but performs badly when b becomes

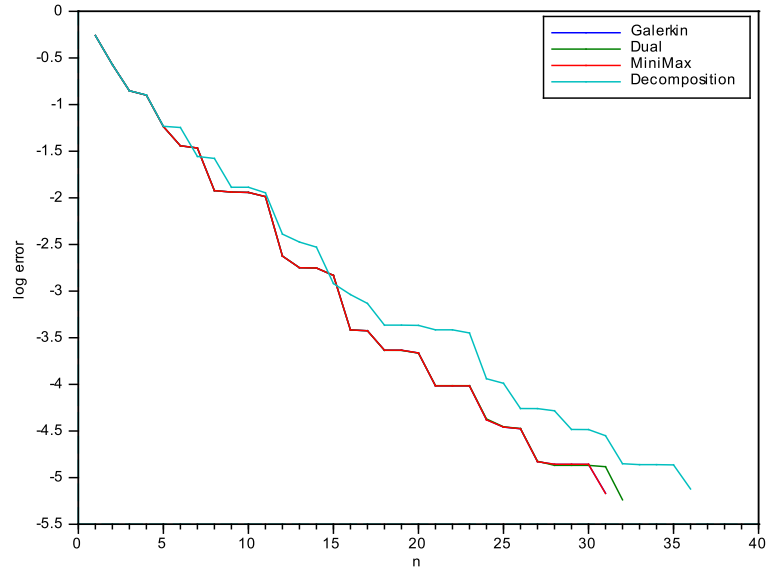


Figure 6: Evolution of $\log_{10} (\|F - D_x U_n - b_x N_x U_n - U\|_{\mathfrak{S}_2})$ for the different algorithms as a function of n with $N = 50$ and $b = 0.01$.

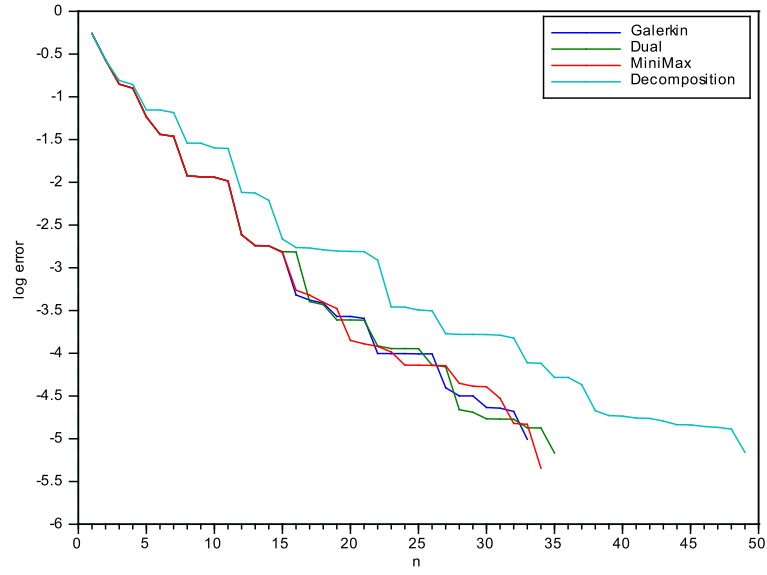


Figure 7: Evolution of $\log_{10} (\|F - D_x U_n - b_x N_x U_n - U\|_{\mathfrak{S}_2})$ for the different algorithms as a function of n with $N = 50$ and $b = 0.1$.

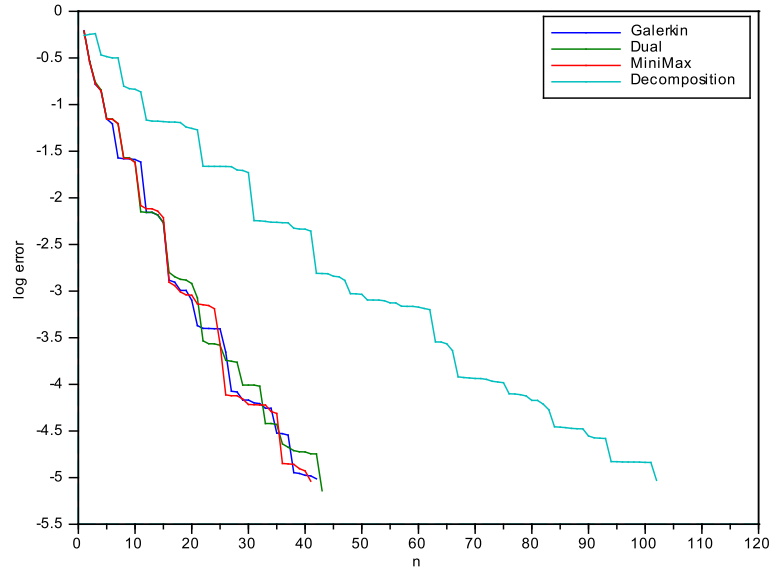


Figure 8: Evolution of $\log_{10} (\|F - D_x U_n - b_x N_x U_n - U\|_{\mathfrak{S}_2})$ for the different algorithms as a function of n with $N = 50$ and $b = 1$.

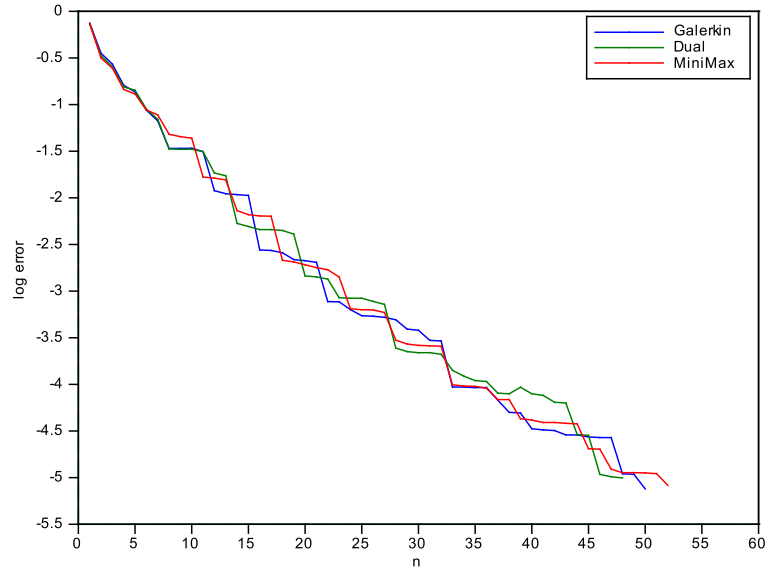


Figure 9: Evolution of $\log_{10} (\|F - D_x U_n - b_x N_x U_n - U\|_{\mathfrak{S}_2})$ for the different algorithms as a function of n with $N = 50$ and $b = 2$.

larger. Let us point out that the CPU time needed to compute one tensor product in the MiniMax or the Dual Greedy algorithms is twice the time needed in the Decomposition or the Galerkin algorithms. Thus, the Decomposition algorithm is more efficient than the Dual or the MiniMax algorithm for small antisymmetric parts of the bilinear form $a(\cdot, \cdot)$ but, when b becomes too large, the algorithm behaves poorly.

7 Appendix: other algorithms

In this section, we present two other possible tracks towards the design of efficient greedy algorithms for high-dimensional non-symmetric problems, for which there is still work in progress.

7.1 An ill-defined (but converging) algorithm

In this section, we first present an algorithm whose iterations are ill-defined in general but for which we can prove the convergence in the full general case. Of course, this algorithm will not be useful in practice but we believe the proof is instructive in our context.

Let $\alpha > 0$. The algorithm reads as follows:

1. set $u_0 = 0$ and $n = 1$;
2. find $(r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in V_x^2 \times V_t^2$ such that

$$\begin{cases} (\tilde{r}_n, \tilde{s}_n) \in \underset{(\tilde{r}, \tilde{s}) \in V_x \times V_t}{\operatorname{argmin}} \frac{1}{2} \|\tilde{r} \otimes \tilde{s}\|_V^2 - l(\tilde{r} \otimes \tilde{s}) - a(u_{n-1} + r_n \otimes s_n, \tilde{r} \otimes \tilde{s}), \\ (r_n, s_n) \in \underset{(r, s) \in V_x \times V_t}{\operatorname{argmin}} \frac{\alpha}{2} \|r \otimes s\|_V^2 - a(r \otimes s, \tilde{r}_n \otimes \tilde{s}_n); \end{cases} \quad (51)$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$.

The Euler equations associated to these coupled minimization problems read: for all $(\delta r, \delta \tilde{r}, \delta s, \delta \tilde{s}) \in V_x^2 \times V_t^2$,

$$\begin{cases} \langle \tilde{r}_n \otimes \tilde{s}_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n \rangle_V = l(\tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) - a(u_{n-1} + r_n \otimes s_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n), \\ \alpha \langle r_n \otimes s_n, r_n \otimes \delta s + \delta r \otimes s_n \rangle_V = a(r_n \otimes \delta s + \delta r \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n). \end{cases} \quad (52)$$

Using Lemma 1.1 and (11), these definitions imply that for all $n \in \mathbb{N}^*$,

$$\|\tilde{r}_n \otimes \tilde{s}_n\|_V = \sup_{(\tilde{r}, \tilde{s}) \in V_x \times V_t} \frac{l(\tilde{r} \otimes \tilde{s}) - a(u_{n-1} + r_n \otimes s_n, \tilde{r} \otimes \tilde{s})}{\|\tilde{r} \otimes \tilde{s}\|_V} \quad (53)$$

and

$$\alpha \|r_n \otimes s_n\|_V = \sup_{(r, s) \in V_x \times V_t} \frac{a(r \otimes s, \tilde{r}_n \otimes \tilde{s}_n)}{\|r \otimes s\|_V}. \quad (54)$$

Indeed, for the first equality, one has to consider the symmetric coercive continuous bilinear form $a^1(\cdot, \cdot) = \langle \cdot, \cdot \rangle_V$ and the continuous linear form $l^1(\cdot) = l(\cdot) - a(u_{n-1} + r_n \otimes s_n, \cdot)$. For the second equality, the symmetric coercive continuous bilinear form to consider is $a^2(\cdot, \cdot) = \alpha \langle \cdot, \cdot \rangle_V$ and the continuous linear form is $l^2(\cdot) = a(\cdot, \tilde{r}_n \otimes \tilde{s}_n)$.

The following result holds:

Proposition 7.1. *Let us assume that the iterations of algorithm (51) are well-defined for all $n \in \mathbb{N}^*$. Then, $(u_n)_{n \in \mathbb{N}^*}$ converges to u in the sense of the injective norm:*

$$\|u - u_n\|_{i\mathcal{A}} = \sup_{(\tilde{r}, \tilde{s}) \in V_x \times V_t} \frac{a(u - u_n, \tilde{r} \otimes \tilde{s})}{\|\tilde{r} \otimes \tilde{s}\|_V} \xrightarrow{n \rightarrow \infty} 0.$$

Proof. Let us first remark that, using (53),

$$\|u - u_n\|_{i\mathcal{A}} = \sup_{(\tilde{r}, \tilde{s}) \in V_x \times V_t} \frac{a(u - u_n, \tilde{r} \otimes \tilde{s})}{\|\tilde{r} \otimes \tilde{s}\|_V} = \|\tilde{r}_n \otimes \tilde{s}_n\|_V.$$

Thus, it is sufficient to prove that the sequence $(\|\tilde{r}_n \otimes \tilde{s}_n\|_V)_{n \in \mathbb{N}^*}$ converges to 0 as n goes to infinity. Let us first prove that this sequence is non-increasing. Let $n \geq 2$. From the Euler equation (52) associated to the minimization problem defining $(\tilde{r}_n, \tilde{s}_n)$, it holds that

$$\|\tilde{r}_n \otimes \tilde{s}_n\|_V^2 = l(\tilde{r}_n \otimes \tilde{s}_n) - a(u_{n-1} + r_n \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n). \quad (55)$$

Besides, using (53) at iteration $n - 1$, we obtain

$$\begin{aligned} \|\tilde{r}_{n-1} \otimes \tilde{s}_{n-1}\|_V &= \sup_{(\tilde{r}, \tilde{s}) \in V_x \times V_t, \tilde{r} \otimes \tilde{s} \neq 0} \frac{l(\tilde{r} \otimes \tilde{s}) - a(u_{n-1}, \tilde{r} \otimes \tilde{s})}{\|\tilde{r} \otimes \tilde{s}\|_V} \\ &\geq \frac{l(\tilde{r}_n \otimes \tilde{s}_n) - a(u_{n-1}, \tilde{r}_n \otimes \tilde{s}_n)}{\|\tilde{r}_n \otimes \tilde{s}_n\|_V}, \end{aligned}$$

which, using (55), leads to

$$\|\tilde{r}_n \otimes \tilde{s}_n\|_V^2 \leq \|\tilde{r}_{n-1} \otimes \tilde{s}_{n-1}\|_V \|\tilde{r}_n \otimes \tilde{s}_n\|_V - a(r_n \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n).$$

Using the second Euler equation (52) defining (r_n, s_n) , we have that $\alpha \|r_n \otimes s_n\|_V^2 = a(r_n \otimes s_n, \tilde{r}_n \otimes \tilde{s}_n)$. Finally, it holds that

$$\|\tilde{r}_n \otimes \tilde{s}_n\|_V^2 \leq \|\tilde{r}_n \otimes \tilde{s}_n\|_V \|\tilde{r}_{n-1} \otimes \tilde{s}_{n-1}\|_V - \alpha \|r_n \otimes s_n\|_V^2. \quad (56)$$

This implies that the sequence $(\|\tilde{r}_n \otimes \tilde{s}_n\|_V)_{n \in \mathbb{N}^*}$ is non-increasing and thus converges towards a limit $z \geq 0$. Let us argue by contradiction and assume that $z > 0$. Dividing by $\|\tilde{r}_n \otimes \tilde{s}_n\|_V$ equation (56), we obtain

$$\alpha \frac{\|r_n \otimes s_n\|_V^2}{\|\tilde{r}_n \otimes \tilde{s}_n\|_V} \leq \|\tilde{r}_{n-1} \otimes \tilde{s}_{n-1}\|_V - \|\tilde{r}_n \otimes \tilde{s}_n\|_V.$$

Since we have assumed that $\|\tilde{r}_n \otimes \tilde{s}_n\|_V \geq z > 0$ for all $n \in \mathbb{N}^*$, the series of general term $(\|r_n \otimes s_n\|_V^2)_{n \in \mathbb{N}^*}$ converges and $\|r_n \otimes s_n\|_V \xrightarrow[n \rightarrow \infty]{} 0$. Using (54), this implies that for all $(r, s) \in V_x \times V_t$,

$$a(r \otimes s, \tilde{r}_n \otimes \tilde{s}_n) \xrightarrow[n \rightarrow \infty]{} 0.$$

Using assumption (A1) and the fact that $(\|\tilde{r}_n \otimes \tilde{s}_n\|_V)_{n \in \mathbb{N}^*}$ is bounded, we have

$$\forall w \in V, a(w, \tilde{r}_n \otimes \tilde{s}_n) \xrightarrow[n \rightarrow \infty]{} 0.$$

Since we have assumed that the operator \mathcal{A} is bijective, it is surjective on V , and the sequence $(\tilde{r}_n \otimes \tilde{s}_n)_{n \in \mathbb{N}^*}$ weakly converges to 0 in V .

Using (53), it holds that for all $n \in \mathbb{N}^*$,

$$\|\tilde{r}_n \otimes \tilde{s}_n\|_V^2 = l(\tilde{r}_n \otimes \tilde{s}_n) - a(u_n, \tilde{r}_n \otimes \tilde{s}_n) = l(\tilde{r}_n \otimes \tilde{s}_n) - a\left(\sum_{k=1}^n r_k \otimes s_k, \tilde{r}_n \otimes \tilde{s}_n\right).$$

Since $(\tilde{r}_n \otimes \tilde{s}_n)_{n \in \mathbb{N}^*}$ weakly converges to 0 in V , necessarily $l(\tilde{r}_n \otimes \tilde{s}_n) \xrightarrow[n \rightarrow \infty]{} 0$.

Besides, using (54), we have

$$\begin{aligned}
\left| a \left(\sum_{k=1}^n r_k \otimes s_k, \tilde{r}_n \otimes \tilde{s}_n \right) \right| &\leq \sum_{k=1}^n |a(r_k \otimes s_k, \tilde{r}_n \otimes \tilde{s}_n)| \\
&\leq \alpha \sum_{k=1}^n \|r_k \otimes s_k\|_V \|r_n \otimes s_n\|_V \\
&\leq \alpha \left(\sum_{k=1}^n \|r_k \otimes s_k\|_V^2 \right)^{1/2} (n \|r_n \otimes s_n\|_V^2)^{1/2}.
\end{aligned}$$

Since the series of general term $(\|r_n \otimes s_n\|_V^2)_{n \in \mathbb{N}^*}$ is convergent, the sequence $(\sum_{k=1}^n \|r_k \otimes s_k\|_V^2)_{n \in \mathbb{N}^*}$ is bounded and there exists a subsequence of $(n \|r_n \otimes s_n\|_V^2)_{n \in \mathbb{N}^*}$ which converges to 0. Thus, there exists a subsequence of $(\|\tilde{r}_n \otimes \tilde{s}_n\|_V)_{n \in \mathbb{N}^*}$ converging to 0 and since the whole sequence converges to z , we have $z = 0$ by uniqueness of the limit. We obtain a contradiction. \square

Remark 7.1. Unfortunately, as announced in the beginning of this section, in general, the iterations of algorithm (51) are not well-defined in the sense that there may not exist a solution $(r_n, \tilde{r}_n, s_n, \tilde{s}_n) \in V_x^2 \times V_t^2$ of the coupled minimization problems. Numerically, we can observe that if we use a coupled fixed-point algorithm similar to the one presented for the Minimax algorithm, the procedure does not converge in general. Finding a suitable way to adapt these ideas in an implementable well-defined algorithm is work in progress.

Remark 7.2. When the bilinear form $a(\cdot, \cdot)$ is symmetric and coercive, all the iterations of algorithm (51) are well-defined. If $\langle \cdot, \cdot \rangle_V$ is chosen to be equal to $a(\cdot, \cdot)$, the second equation of (52) implies that $r_n \otimes s_n = \frac{1}{\alpha} \tilde{r}_n \otimes \tilde{s}_n$ and the first equation of (52) can be rewritten as: for all $(\delta \tilde{r}, \delta \tilde{s}) \in \tilde{V}_x \times \tilde{V}_t$,

$$\left(1 + \frac{1}{\alpha} \right) \langle \tilde{r}_n \otimes \tilde{s}_n, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n \rangle_V = l(\tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n) - \langle u_{n-1}, \tilde{r}_n \otimes \delta \tilde{s} + \delta \tilde{r} \otimes \tilde{s}_n \rangle_V.$$

This Euler equation is similar to the Euler equation of the first iteration of the standard greedy algorithm applied to the symmetric coercive problem:

$$\begin{cases} \text{find } \tilde{u} \in V \text{ such that} \\ \forall v \in V, \left(1 + \frac{1}{\alpha} \right) \langle \tilde{u}, v \rangle_V = l(v) - \langle u_{n-1}, v \rangle_V. \end{cases}$$

Thus, if we consider now the following (well-defined) algorithm

1. set $u_0 = 0$ and $n = 1$;
2. find $(r_n, s_n) \in V_x \times V_t$ such that

$$(r_n, s_n) \in \underset{(r,s) \in V_x \times V_t}{\operatorname{argmin}} \frac{\lambda}{2} \|r \otimes s\|_V^2 - l(r \otimes s) - \langle u_{n-1}, v \rangle_V;$$

3. set $u_n = u_{n-1} + r_n \otimes s_n$ and $n = n + 1$,

following the proof of Proposition 7.1, we can prove that the sequence $(u_n)_{n \in \mathbb{N}^*}$ converges to \mathcal{L} in the sense of the injective norm as soon as $\lambda > 1$. Let us point out that when $\lambda = 1$, this algorithm is identical to the standard greedy algorithm applied to the symmetric coercive problem

$$\begin{cases} \text{find } \tilde{u} \in V \text{ such that} \\ \forall v \in V, \langle \tilde{u}, v \rangle_V = l(v). \end{cases}$$

7.2 Link with a symmetric formulation

Let us now present another approach, for which no convergence result have been proved so far. The idea is based on the article [7] by Cohen, Dahmen and Welper, where the objective was to develop stable formulations of multiscale convection-diffusion equations.

The principle of the method is to reformulate the antisymmetric problem (16) defined on the Hilbert space V , as a symmetric problem defined on the Hilbert space $V \times V$. Indeed, it is proved in [7] that the unique solution of the problem

$$\begin{cases} \text{find } (v, \tilde{v}) \in V \times V \text{ such that} \\ a(w, \tilde{v}) = 0, \quad \forall w \in V, \\ a(v, \tilde{w}) - \langle R_V \tilde{v}, \tilde{w} \rangle = l(\tilde{w}), \quad \forall \tilde{w} \in V, \end{cases} \quad (57)$$

is $(v, \tilde{v}) = (u, 0)$ where u is the unique solution of (16).

This new problem is now *symmetric*. It is equivalent to the following problem

$$\begin{cases} \text{find } (v, \tilde{v}) \in V \times V \text{ such that} \\ \begin{pmatrix} 0 & A^* \\ A & -R_V \end{pmatrix} \begin{pmatrix} v \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} 0 \\ L \end{pmatrix} \text{ in } V' \times V'. \end{cases}$$

This new formulation of the problem is symmetric, but *not coercive*. No convergence results exist for greedy methods in this framework. However,

the situation seems more encouraging than in the original non-symmetric case. It is to be noted though that the use of a simple Galerkin algorithm, similar to the one introduced in Section 2.3, does not work in this case either. More subtle algorithms need to be designed in this case as well, and this is currently work in progress.

References

- [1] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics*, 139:153–176, 2006.
- [2] R.E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [3] G. Bonithon and A. Nouy. Tensor-based methods and proper generalized decompositions for the numerical solution of high-dimensional problems: alternative definitions. *preprint*, hal.archives-ouvertes.fr/hal-00664061, 2012.
- [4] A. Buffa, Y. Maday, A.T. Patera, C. Prud’homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:595–603, 2012.
- [5] E Cancès, V. Ehrlacher, and T. Lelièvre. Convergence of a greedy algorithm for high-dimensional convex problems. *Mathematical Models and Methods in Applied Sciences*, 21:2433–2467, 2011.
- [6] A. Chkifa, A. Cohen, R. DeVore, and C. Schwab. Sparse Adaptive Taylor Approximation Algorithms for Parametric and Stochastic Elliptic PDEs. *Submitted*, 2011.
- [7] A. Cohen, W. Dahmen, and G. Welper. Adaptivity and variational stabilization for convection-diffusion equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:1247–1273, 2012.
- [8] J. Diestel and J. Fourie, J.H. and Swart. *The Metric Theory of Tensor Products*. American Mathematical Society, 2008.

- [9] A. Falco, L. Hilario, N. Montès, and M.C. Mora. Numerical strategies for the Galerkin-Proper Generalized Decomposition Method. *Mathematical and Computer Modelling (in press)*, 2012.
- [10] L. Figuerola and E. Suli. Greedy Approximation of High-Dimensional Ornstein-Uhlenbeck Operators. *Foundations of Computational Mathematics*, 12:573–623, 2012.
- [11] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer, 2012.
- [12] P. Ladevèze. *Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation*. Springer, Berlin, 1999.
- [13] C. Le Bris, T. Lelièvre, and Y. Maday. Results and Questions on a Non-linear Approximation Approach for Solving High-dimensional Partial Differential Equations. *Constructive Approximation*, 30:621–651, 2009.
- [14] A. Lozinski. Interplay between PGD and greedy approximation algorithms for high-dimensional PDEs: symmetric and not symmetric cases. <http://website.ec-nantes.fr/reduc/>, 2010.
- [15] A. Nouy. Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations. *Archives of Computational Methods in Engineering*, 16:251–285, 2009.
- [16] A. Nouy. A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199:1603–1626, 2010.
- [17] A. Nouy. A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199:1603–1626, 2010.
- [18] A. Nouy and A. Falco. Proper Generalized Decomposition for Nonlinear Convex Problems in Tensor Banach Spaces. *Numerische Mathematik*, 121:503–530, 2012.

- [19] L. Skrzypek. Uniqueness of Minimal Projections in Smooth Matrix Spaces. *Journal of Approximation Theory*, 107:315–336, 2000.
- [20] V.N. Temlyakov. Greedy Approximation. *Acta Numerica*, 17:235–409, 2008.
- [21] T. von Petersdorff and C. Schwab. Numerical solution of parabolic equations in high dimensions. *M2AN Mathematical Modelling and Numerical Analysis*, 38:93–127, 2004.